

CA Ideal™ for CA Datacom®

Command Reference Guide
Version 14.02



This Documentation, which includes embedded help systems and electronically distributed materials (hereinafter referred to as the "Documentation"), is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2015 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA products:

- CA Datacom®/DB
- CA Datacom® CICS Services
- CA Ideal™ for Datacom® (CA Ideal)
- CA Ideal™ for DB2
- CA IPC

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

Contents

Chapter 1: Preliminary Concepts	15
Command Overview	15
Notation Conventions	15
Defaults	17
Defaults That Cannot Be Changed	17
Defaults Set for an Entire Site	17
Defaults Set for an Individual Session	18
Using Abbreviations for Commands	19
Standard Abbreviations	19
Alternate Abbreviations	19
Abbreviation Exceptions	20
Allowable Abbreviations	20
Summary of Command Abbreviations	20
Using Version Clauses in Commands	23
Using String Delimiters in Commands	26
Using an Asterisk to Represent the Current Entity	27
 Chapter 2: CA Ideal Commands	 27
ALTER MEMBER Command	27
Operand Definitions	28
ALTER OUTPUT Command	28
Operand Definitions	28
ALTER OUTPUT DESTINATION Command	29
Operand Definitions	30
ALTER OUTPUT DESTINATION ALL DISPOSITION Command	31
Operand Definitions	31
ALTER PANEL RUN-STATUS Command	31
Operand Definitions	31
ALTER PROGRAM DATAVIEW Command (CA Datacom CBS Access)	32
Operand Definitions	32
ALTER PROGRAM DATAVIEW Command (VSE)	33
Operand Definitions	33
ALTER PROGRAM DBID Command (CA Datacom CBS Access)	34
Operand Definitions	34
ALTER PROGRAM ENVIRONMENT Command (CA Datacom SQL Access)	35
Operand Definitions	35

ALTER PROGRAM RUN-STATUS Command	38
ALTER SIGNON Command	38
Operand Definitions	38
ASSIGN AUTHORIZATION Command (SQL Access)	39
Operand Definitions	39
DB2 Considerations	41
ASSIGN DATAVIEW Command (CA Datacom Native Access)	41
Operand Definitions	41
ASSIGN DATAVIEW Command (VSE Only)	42
Operand Definitions	42
ASSIGN DBID Command (CA Datacom Native Access)	42
Operand Definitions	42
ASSIGN PROGRAM Command	43
Operand Definitions	43
ASSIGN REPORT Command	44
Operand Definitions	44
BLANKFILL Command	45
CATALOG DATAVIEW Command	46
Operand Definitions	47
CLARIFY Command	49
COMBINE Command	49
Comment.....	50
COMPILE Command	50
Operand Definitions	51
COPY Command	53
Operand Definitions	53
CREATE DATAVIEW Command	54
Operand Definitions	55
CREATE MEMBER Command	55
Operand Definitions	55
CREATE MODULE Command	56
Operand Definitions	56
CREATE PACKAGE Command (DB2 Only)	58
Operand Definitions	58
CREATE PANEL Command	58
Operand Definitions	59
Using Dataview with the CREATE Command	59
CREATE PLAN Command (DB2 Only)	60
Operand Definitions	60
CREATE PROGRAM Command.....	61
Operand Definitions	61
CREATE REPORT Command	62

Operand Definitions	62
CREATE SYSTEM Command	63
Operand Definitions	63
CREATE USER Command	63
Operand Definitions	64
DATAQUERY Command	64
Operand Definitions	64
DATAVIEW Command	65
DEBUG Command	65
Operand Definitions	66
DEFINE AUTHORIZATION Command (CA Datacom SQL Access)	68
Operand Definitions	69
DEFINE OUTPUT DESTINATION NETWORK Command	69
Operand Definitions	70
DEFINE OUTPUT DESTINATION SYSTEM Command	71
Operand Definitions	71
DELETE Command	72
Operand Definitions	72
DELETE AUTHORIZATION Command (CA Datacom SQL Access)	73
Operand Definitions	73
DELETE MEMBER Command	73
Operand Definitions	73
DELETE MODULE Command	74
Operand Definitions	74
DELETE OUTPUT Command	75
Operand Definitions	75
DELETE OUTPUT DESTINATION Command	75
Operand Definitions	75
DELETE PACKAGE Command (DB2 Only)	76
Operand Definitions	76
DELETE PLAN Command (DB2 Only)	76
Operand Definitions	76
DEQUEUE Command	77
User Considerations	78
DETAIL Command	78
DISABLE Command	79
Operand Definitions	79
Important User Considerations	81
DISPLAY AUTHORIZATION Command	81
DISPLAY DATAVIEW Command	82
Operand Definitions	82
DISPLAY INDEX Command	83

Operand Definitions	84
Displaying Related Programs	86
Displaying Related Dataviews	88
DISPLAY INDEX Line Commands	89
Processing Hierarchy	90
Error Processing	93
DISPLAY INDEX MODULE Command	93
Operand Definitions	94
DISPLAY INDEX OUTPUT Command	94
DISPLAY INDEX SESSION Command	95
Field Definitions	96
DISPLAY JOBCARD Command	96
Operand Definitions	97
DISPLAY LIBRARY STATUS Command	97
DISPLAY LMT Command	97
DISPLAY MEMBER Command	98
Operand Definitions	99
DISPLAY OUTPUT Command	99
Operand Definitions	99
DISPLAY OUTPUT DESTINATION Command	100
Field Definitions	100
DISPLAY OUTPUT STATUS Command	101
Operand Definitions	101
DISPLAY PACKAGE Command (DB2 Only)	105
Operand Definitions	105
DISPLAY PANEL Command	105
Operand Definitions	106
DISPLAY PCT Command	107
DISPLAY PLAN Command (DB2 Only)	109
Operand Definitions	109
DISPLAY PROGRAM Command	110
Operand Definitions	110
DISPLAY REPORT Command	111
Operand Definitions	111
DISPLAY SESSION OPTIONS Command	112
Operand Definitions	112
DISPLAY SYSTEM Command	113
Operand Definitions	113
DISPLAY USER Command	113
Operand Definitions	114
DUPLICATE Command	114
Operand Definitions	115

DUPLICATE MEMBER Command	116
Operand Definitions	116
DUPLICATE PACKAGE Command (DB2 Only)	117
Operand Definitions	117
DUPLICATE PLAN Command (DB2 Only)	118
Operand Definitions	118
EDIT DATAVIEW Command	118
Operand Definitions	119
EDIT JOBCARD Command	119
Operand Definitions	120
EDIT MEMBER Command	120
Operand Definitions	120
EDIT PACKAGE Command (DB2 Only)	121
Operand Definitions	121
EDIT PANEL Command	121
Operand Definitions	122
EDIT PLAN Command (DB2 Only)	123
Operand Definitions	123
EDIT PROGRAM Command	123
Operand Definitions	124
EDIT REPORT Command	125
Operand Definitions	126
EDIT SYSTEM Command	126
Operand Definitions	127
EDIT USER Command	127
Operand Definitions	127
ENABLE RUN Command	128
Operand Definitions	128
ENVIRONMENT Command (CA Datacom SQL Access)	129
EXECUTE Command	130
Operand Definitions	130
GENERATE PACKAGE Command (Batch, DB2 Only)	131
Operand Definitions	131
GENERATE PLAN Command (Batch, DB2 Only)	132
Operand Definitions	132
HELP Command	133
Operand Definitions	133
IDENTIFICATION Command	134
IDENTIFY MODULE Command (Batch Only)	134
Operand Definitions	135
IF ELSE ENDIF Commands	135
Operand Definitions	135

IRULES Command	136
KEEP OUTPUT Command.....	136
LAYOUT Command	136
Operand Definitions	137
MARK STATUS Command	137
Operand Definitions	138
NULLFILL Command	138
OFF Command.....	138
OFFON Command.....	139
ORULES Command	139
PARAMETER Command	139
PRINT Command	140
Operand Definitions	140
PRINT INDEX Command	142
Operand Definitions	143
Printing Program Relationships.....	145
PRINT INDEX MODULE Command	148
Operand Definitions	148
PRINT INDEX OUTPUT Command	148
PRINT INDEX SESSION Command	149
Field Definitions	149
CICS Considerations	150
PRINT MEMBER Command	150
Operand Definitions	151
PRINT OUTPUT DESTINATION Command	151
PRINT OUTPUT STATUS Command.....	152
Operand Definitions	152
PRINT PACKAGE Command (DB2 Only)	153
Operand Definitions	153
PRINT PLAN Command (DB2 Only).....	154
Operand Definitions	154
PRINT SCREEN Command	155
Operand Definitions	155
PRINT SESSION OPTIONS Command	156
Operand Definitions	156
PROCEDURE Command	156
PROCESS Command.....	157
PRODUCE REPORT Command	157
Operand Definitions	158
Other Program Requirements.....	159
PROGRAM Command	159
PURGE Command	159

Operand Definitions	160
REBIND Command (CA Datacom SQL Access)	160
Operand Definitions	160
REFORMAT Command	161
Operand Definitions	161
REFRESH Command	162
Programs and Panels	162
Plans	162
Other Program Requirements	163
REPORT Command	163
RESET Command	164
Operand Definitions	164
RESOURCES Command	165
RUN Command	165
Operand Definitions	166
SELECT SYSTEM Command	168
Operand Definitions	169
SET \$RETURNCODE Command	169
Operand Definitions	169
SET ASYNCMMSG Command	169
Operand Definitions	169
SET CATALOG VALIDATION Command	170
Operand Definitions	170
SET SITE CHECK DUPLICATE Command	170
Operand Definitions	171
SET COMMAND Command	171
Operand Definitions	172
SET COMPILE Command	174
Operand Definitions	175
SET DATAVIEW VERSION Command (NonSQL Dataviews)	177
Operand Definitions	177
SET DBSQL Command (CA Datacom SQL Access)	178
Operand Definitions	179
SET EDIT Command	181
Operand Definitions	181
SET ENVIRONMENT Command	183
Operand Definitions	184
Other Plan Considerations	188
SET OUTPUT Command	188
Operand Definitions	189
SET PANEL Command	191
Operand Definitions	193

SET PANEL SESSION OPTIONS Command	203
Panel Session Options Considerations	204
SET PLAN MAXSQL Command (DB2 Only)	204
Operand Definitions	204
SET REPORT Command	205
Operand Definitions	205
SET RUN Command	208
Operand Definitions	209
SET SCROLL Command	212
Operand Definitions	212
SET SITE Commands	213
Edit Options	213
SET VERSION Command	217
Operand Definitions	217
SIGNON Command	217
SPLIT Command	218
Operand Definitions	218
SUBMIT Command	220
Operand Definitions	220
SYSTEM Command	221
TIME Command	221
USER Command	221
WORK Command	222

Chapter 3: Editing in CA Ideal 223

Editing in CA Ideal	223
Editing Using 3270 Hardware Facilities	229
Editing Using Primary Editing Commands	229
Editing Using Line Commands	230
Order of Command Processing	233
CHANGE Command	234
Operand Definitions	234
CHECKPOINT Command	240
COPY Command	240
Operand Definitions	241
Copy Line Command	242
Operand Definitions	243
DELETE Command	245
Operand Definitions	246
Delete Line Command	246
Operand Definitions	246

Operand Definitions	248
DISPLAY ROLLBACK Command	249
EXCLUDE Command	249
Operand Definitions	250
Components Used with the EXCLUDE Command	251
FIND Command	253
Operand Definitions	253
Components Used with the FIND Command	255
FIRST Command	259
Operand Definitions	259
Components Used with the FIRST Command	260
IGNORE Line Command	262
INCLUDE Command	263
Operand Definitions	263
INPUT Command	266
Input Line Command	268
LAST Command	273
MOVE Command	275
Move Line Command	276
NEXT Command	279
POSITION Command	281
PREVIOUS Command	282
RENUMBER Command	285
Repeat Line Command	287
RESET Command	290
RESET Line Command	290
RESHOW Line Command	291
ROLLBACK Command	293
Scroll Command	293
Vertical Scrolling	294
Horizontal Scrolling	295
Scroll Line Command	296
SHIFT Command	299
Shift Line Command	301
Templates	302

Chapter 1: Preliminary Concepts

This guide describes the commands used in the CA Ideal environment. It includes the syntax and a description of each command. The commands are divided into two sections:

- *Primary commands*, such as OFF and DISPLAY, are entered on the command line or executed from a member.
- *Editing commands*, such as MOVE and CHANGE, are entered on the command line of a display being edited. Editing line commands, such as CC and MM, are entered in the sequence number area of a display being edited.

Command Overview

The commands are presented in alphabetical sequence in two sections for easy reference. With this guide, you should be able to find complete information on any CA Ideal command quickly and easily.

This chapter describes:

- The command notation used in this guide.
- How defaults for commands are handled in CA Ideal and how they are documented in this guide.
- Rules for abbreviating CA Ideal commands, with a list of exceptions.
- Rules for using version clauses in CA Ideal commands.
- Rules for using string delimiters in CA Ideal commands.

For basic information on what CA Ideal is and how to use it, see the *Working in the Environment Guide*.

Notation Conventions

This guide uses the following rules and special characters in syntax illustrations.

Enter exactly as shown in command syntax:

Notation	Description of use
UPPERCASE	Identifies commands, keywords, and keyword values that you must enter exactly as shown or replace it with an authorized abbreviation.

Notation	Description of use
symbols	You must enter all special characters, such as parentheses and quotation marks (but not ellipses, brackets, and braces) as shown.

Do not enter the following as shown; notations clarify command syntax:

Notation	Description of use
lower case italics	Represent a value you must supply.
Brackets []	Identifies optional keywords or clauses, or a group of options from which, if included at all, you must make a choice.
Braces { }	Enter one of the keywords or clauses.
Underlining	Indicates a CA Ideal default that you cannot change with a SET command and, therefore, you do not need to specify.
Ellipses ...	You can repeat the preceding word or clause.

Note: For more information about the use of special characters in names, see the Concepts section of the *Programming Reference Guide*.

Example

This example shows syntax format of CA Ideal commands.

```
COMmand keyword1={AAA} [keyword2] [keyword3={CCC }]  
                {BBB}          [          {xxxxxxxx}]
```

COM

Initiates the command. COM is an abbreviation.

keyword1=

Designates a required keyword followed by a choice of required values. You must specify one and only one of these values.

keyword2

Specifies an optional keyword.

keyword3=

Designates an optional keyword followed by a choice of values, one of which is required if you specify the parameter.

CCC

Designates a default value for keyword 3. If you do not specify the keyword, the product uses this value.

XXXXXXXX

Indicates that you can specify a value according to the instructions in the keyword description following the syntax diagram.

Defaults

When this document references defaults, in most cases no actual default value is mentioned. This is because the defaults that CA Ideal uses are established in a number of ways and are often specific to the individual user.

CA Ideal is delivered with default values for all options. You can change some of these defaults and cannot change others.

Defaults That Cannot Be Changed

Some defaults are for certain choices in CA Ideal commands and fill-ins, and cannot be changed. This type of default is underlined in the command syntax examples in this guide. An example of this type of default is the following command, which defaults to the procedure component of the program definition:

```

                                [ ENVIRONMENT ]
                                [ RESOURCES   ]
EDIT PROGRAM name [ PARAMETER   ]
                                [ WORK       ]
                                [ PROCEDURE   ]

```

Defaults Set for an Entire Site

You can establish defaults for the entire site. You can change some site defaults only during installation. You can reset them later only by rerunning installation jobs. An example of a default in this category is a default library name.

You can reset other site defaults by using either the SET SITE command or the fill-ins provided for setting certain session control and print options. Any default that is reset with either a SET SITE command or a site options fill-in becomes a new site default that remains in effect unless it is reset with another SET SITE command or fill-in.

For more information about SET SITE commands, see the *Administration Guide*.

Defaults Set for an Individual Session

Each CA Ideal user can set defaults for an individual session using SET commands or session options fill-ins.

A default set with a SET command or session options fill-in is changed only for the user who issued the command. It remains in effect only for the current session. The user can store SET commands in a member called SIGNON. This executes the commands automatically each time the user signs on. It works as if that user's default settings were changed permanently.

References to defaults in this document refer to whatever default is currently in effect for any given option. The actual default for any given option, for any given user, in any given session, at any given site, depends on what choices for setting defaults were made.

You can display or print the default option settings for the current session by using the following commands:

```
DISPLAY SESSION OPTIONS
PRINT SESSION OPTIONS
```

Following are the default values for the CA Ideal SET commands. They are the defaults that are delivered with CA Ideal. As explained above, these defaults might be changed at your site.

Command	Default
SET [SITE] CATALOG VALIDATION	NO
SET [SITE] COMPILE LINELIMIT	999999
SET [SITE] DBSQL DECPOINT	P
SET [SITE] DBSQL OPTMSG	NON
SET [SITE] DBSQL STRDELIM	A
SET [SITE] EDIT COLUMNS	MAX
SET [SITE] ENVIRONMENT FINAL-ID	NONE
SET [SITE] PANEL ERRORHANDLING	B
SET [SITE] PANEL IFATTRIBUTE	UAL
SET [SITE] PANEL INFILL	0
SET [SITE] PANEL TFATTRIBUTE	PSL
SET [SITE] RUN CBSTRACE	NO
SET [SITE] RUN PLAN	IDP110DV

Command	Default
SET [SITE] RUN QUITIDEAL	NO
SET [SITE] RUN SQL	MIXED
SET [SITE] RUN STRNO	1
SET [SITE] RUN XA	OFF

Using Abbreviations for Commands

This section lists abbreviations for commands used in CA-Ideal.

Standard Abbreviations

The standard abbreviation for a command or command option is the first three characters of the word. For example, the standard abbreviation for the COMPILER command is COM. All SCF commands (SET CMD xxx) check only the first three characters of the operand following the SET CMD. Anything entered beyond that point is ignored.

Alternate Abbreviations

For some commands and options, CA Ideal also accepts alternate abbreviations. For example, in addition to BAC, the abbreviations BACK and BWD are accepted for BACKWARD.

Command/Option	Exception	Alternate
BACKWARD	(none)	BACK, BWD
HEADER	HDR	
POSTMSG	MSG	
TRAILER	TRL	TRLR

Abbreviation Exceptions

There are a number of exceptions to the standard first three-character abbreviation. These exceptions are abbreviations for command words whose first three characters are not unique and, therefore, would conflict with an abbreviation for another command. Other abbreviations (or no abbreviation at all) are used.

For example, the COMBINE command does not have a three-character abbreviation. The COMMAND option accepts the abbreviation CMD only:

Command/Option	Exception	Alternate
COMBINE	COMBINE	(none)
COMMAND	CMD	(none)

Allowable Abbreviations

CA Ideal keywords are only validated to the point where it can be determined that they are unique. Anything entered beyond that point is ignored.

For example, the command SET SITE ASYNCMSG recognizes the values N for NONE and U for USER. However, values such as NO, NON, NOOOO... are also accepted for NONE, since it was already determined from just entering "N" what the value really is. Everything entered beyond that point is disregarded. The same holds true for USER. Since "U" was already identified as USER, values such as US, USE, USEEEE... are also acceptable.

Summary of Command Abbreviations

Because most abbreviations are standard, they are not shown in the syntax illustrations in this guide. The following list shows command abbreviations that are either exceptions to the standard abbreviation or are alternatives available in addition to the standard abbreviation.

Command/Option	Exception	Alternate
AUTHORIZATION		AUTH, ATZ
BACKWARD		BACK, BWD
BLANKFILL		BLK, BLANK, BLANKS
BOTTOM		BOT
CATALOG		CATLG

Command/Option	Exception	Alternate
CHANGE		C
COLUMNS		COLUMN
COLVIEW		COL
COMBINE	COMBINE	
COMMAND	CMD	
COMMENT	COMMENT	
CURSOR-STABILITY	CS	
DATAQUERY	DQ	DQRY
DATAVIEW		DVW
DATEFORM	DATE	
DEBUG		DEBUG
DELIMITER	DLM	DELIM
DESCRIPTION	DESC	
DESTINATION	DEST	
DISPLAY		D
DISPOSITION	DISP	
EDIT		E
EXCLUDE		EXCL, EXCLUDING
EXECUTE		EXEC
EXECTIME	EXEC	
EXPORT		
FIELD		FLD
FIND		F
FORWARD		FWD
FULL		F
HEADER	HDR	
HELP	HELP	
HISTORY		HIST
HORIZONTAL		HOR
IDENTIFICATION		IDENT

Command/Option	Exception	Alternate
IMPORT		IMP
INCLUDE		INCL, INCLUDING
ISOLATION-LEVEL		ISOLATION
LEFT (in SET EDIT)		L
LINES		LINE
MAIL	MAIL	
MESSAGE		MSG
NAMES		NAME
NO		N
NOHEADING		NO
NOVALIDATION		NOVAL, NOVALID
NULLFILL		NUL, NULLS, NULLFIL
OFF	OFF	N, NO
ON	ON	Y, YES
OPTIONS		OPTION
PACKAGE	PACKAGE, PKG	
PACKLIST	PACKLIST, PKL	
PAGE		PAGES
PANEL		PNL
PARAMETERS		PARAMETER, PARM, PARMS
PASSWORD		PSW
POSTMSG	MSG	
PREPTIME	PREP	
PREVIOUS		PREV
PROCEDURE	PROC	
PROCESS	PROCESS	
PRODUCTION	PROD	
PROGRAM	PROG	PGM
REPEAT	REPEAT	

Command/Option	Exception	Alternate
REPORT		RPT
RESHOW	RESHOW	
RIGHT (in SET EDIT)		R
ROLLBACK		ROLL
SCREEN	SCREEN	
SHORT		S
SIDEVIEW		SIDE
TRAILER	TRL	TRLR
USER		U, USR
VALIDATION		VALID
VERIFICATION		VERIFY
VERSION		V
VERTICAL		VER
WAITTIME	WAIT	
WIDEOPTION	WIDE	

Using Version Clauses in Commands

There are some rules that apply when specifying version clauses in CA Ideal commands. Following is a description on version clauses and how to use them.

For SQL Dataviews

Only one version is assigned; version 1 in production status. CA Ideal generally does not display this version number or status or require that the user specify it.

For All Entity Types Except Modeled and SQL Dataviews

Each named definition of a given type can exist in one or more forms, called versions, each of which is identified by a number assigned by CA Ideal. There can be as many as 999 versions with the same name. CA Ideal assigns numbers to versions sequentially as they are created, starting with number 1. The application developer cannot modify version numbers.

For CA Datacom/DB Dataviews and Sequential File Dataviews Modeled in the Dictionary

CA Ideal assigns numbers to test versions separately from production and history versions. Test versions are identified as T1 through T999. Production and history versions are identified as 1 through 999.

Editing a definition has no effect on the version number. No matter how many changes are made, the version number remains the same. You can only create new versions with the same name by using the DUPLICATE...NEXT VERSION command. This command makes a copy of an existing version. CA Ideal assigns each new version of a definition the next higher number than the highest previously assigned number. When a version is deleted from the system, unless it was the highest, its number is never reassigned to another version with the same name.

Each definition is uniquely identified by the combination of its type, name, and version number, and can always be referenced by this combination.

There are three cases when you can reference a particular definition without using the version number:

- You can reference the production status version of a definition by replacing the version number with the term **PROD** (since there can be only one production version at a time).
- For any type of definition except for an SQL dataview definition, you can reference the most recently created version of a definition by replacing the version number with the term **LAST**.
- The version clause is optional when specifying **PROD** or **LAST**. For example, if the production version of a report definition named **SALARIES** is version 5, you can reference it as **REPORT SALARIES VERSION 5** or as **REPORT SALARIES VERSION PROD**. If there are seven versions of a program named **UPDATE** (numbered 1 through 7), you can reference the most recently created version as **PROGRAM UPDATE VERSION 7** or as **PROGRAM UPDATE VERSION LAST**.

The following is a description about usage of version clauses:

Status

Each version is assigned to a category that is based on the stage it reached in the production process. This category is called the status of the version. A version can be in test status, production status, or history status. Except for dataview entity types, you can only change the status of a version with the MARK STATUS command.

The following rules apply to each type of status:

Test

- **For all entity types except modeled and SQL dataviews:** When a version is created in CA Ideal by either the CREATE or DUPLICATE command, the version is in test status. You can modify a version as long as it remains in test status. There can be many versions in test status at one time. If a dataview is edited, you must recatalog it before it can be reused.
- **For CA Ideal dataviews and sequential file dataviews modeled in the dictionary:** Test versions are created and maintained in the dictionary. There can be up to 999 test versions at one time.
- **SQL dataviews** can not be in test status.

Production

When a version is created, edited, and tested, and is ready for use in an application, it is marked to production and becomes the production version. Only one version of a definition can be in production status at a time.

A production application must consist of components that are also in production status. This protects its integrity.

- **For all entity types except modeled and SQL dataviews:** You cannot edit or delete the production version. If a production version of a program is compiled, a compilation listing is produced; a new object program is not created.
- **For SQL dataviews:** There can be only one version of a definition at a time, version 1 in production status. SQL dataviews are created through the CA Ideal CATALOG command. You can delete them if they are resources of production programs. You cannot edit them.

History

History versions of a definition are former production versions of that definition.

- **For all entity types except SQL dataviews:** Marking a test version to production automatically retires the existing production version, if any, to history status.
- **SQL dataviews** can not be in history status.

The maximum number of history status versions that you can save is recorded in the dictionary (as the ENTY-HIST-VERS attribute of the TABLE entity in the DATA-DICT database). When this number is exceeded, the oldest history versions are automatically deleted. You can modify the installed default of three history versions.

Using String Delimiters in Commands

The character used to delimit strings in CA Ideal commands is shown as either an apostrophe (') or slash (/). You can use any special character as a string delimiter in a command except:

- The currently defined command delimiter; semicolon (;) is the installed default.
- The currently defined comment character; colon (:) is the installed default.
- Asterisk (*).

National characters - In the USA, these are:

- At sign (@).
- Pound sign (#).
- Dollar sign (\$).

You must use the same character consistently to delimit strings in a command.

To display the currently defined command delimiter character and comment character, use the DISPLAY SESSION OPTIONS command.

Using an Asterisk to Represent the Current Entity

Many CA Ideal commands allow the use of an asterisk (*) to represent the current entity. For example, after editing a program, the command

```
COMPILE *
```

will compile the same program.

The current entity is shown at the top of the panel.

```

=>
=>
=>

-----
DIDA : PROCEDURE DEFINITION  PGM SCROLLER (003) PROD          SYS: $ID  DISPLAY
Command.....1.....2.....3.....4.....5.....6.....7..
===== T O P =====
000010 <<SCROLLER>> PROC
000020 :=====;
000030 : :
000040 : Display all ITEM records and allow their update. :
000050 : :
000060 : This sample program is designed to show how to scroll data :

```

Chapter 2: CA Ideal Commands

The commands covered in this chapter are in alphabetical sequence. A brief description, command syntax, operand definitions, and examples are also provided for easy reference.

ALTER MEMBER Command

The ALTER MEMBER command changes the description of a member.

This command has the following format:

```

{ *                }
ALTER {MEMBER name USER {id }}DESCRIPTION 'string'
{                {name}}

```

Operand Definitions

Operand	Description
*	Use an asterisk as a substitute for MEMBER name if necessary.
<i>name</i>	One- to eight-character member name.
USER <i>clause</i>	CA Ideal user name or one- to three-character user ID associated with the specified member.
DESCRIPTION ' <i>string</i> '	New 1- to 24-character description of the member. For a description of valid string delimiters, see the "Preliminary Concepts" chapter.

Example

```
ALTER MEMBER JCLDEMO DESCRIPTION 'DEMONSTRATION'
```

ALTER OUTPUT Command

The ALTER OUTPUT command changes the output destination, disposition, number of copies, or the retention period for outputs residing in the output library.

This command has the following format:

```

                                {DESTINATION dest}
      {name } {DISPOSITION disp}
ALTER OUTPUT {number} {COPIES nn }
                                {RETENTION value }
```

Operand Definitions

Operand	Description
<i>name</i>	One- to eight-character member name of the output to alter.
<i>number</i>	One- to four-digit output number.
DESTINATION <i>dest</i>	Changes the destination of the output to: LIBRARY The output library. SYSTEM name The system printer name. NETWORK name The network printer.

DISPOSITION <i>disp</i>	Changes the disposition of the output to HOLD, RELEASE, READY, or KEEP as specified.
COPIES <i>nn</i>	Changes the number of copies of output requested. This can be any number from 1-99. This clause is ignored in batch.
RETENTION <i>value</i>	Changes the number of days the output is retained. The value can be any number of days from one up to the site maximum.

ALTER OUTPUT DESTINATION Command

The ALTER OUTPUT DESTINATION command alters the default settings for the specified printer. These values override the actual printer settings. You can define a system type output destination to CA Ideal that is actually a data set. You can then direct CA Ideal reports to that data set through the ASSIGN REPORT command in a batch run. For example:

```
...
//arpt DD DSN=site.dataset.b1330,DISP=OLD
//SYSIN *
ASSIGN REPORT arpt TO DEST SYS b1330
RUN apgm
...
```

B1330 is defined to CA Ideal as a SYStem OUTput DESTination with a blocksize specified.

This command has the following format:

```
ALTER OUTPUT DESTINATION dest {WIDTH width }
                                {HEADER opt }
                                {TRAILER opt } ...
                                {FF opt }
                                {PERTASK num }
                                {BLKSIZE nnnnn }
                                {DISPOSITION disp}
```

Note: The syntax is positional. This allows you to specify multiple options, but you must specify them in the same order as shown in the previous display. For example, ALT OUT DEST xxxx HDR NO TRL NO is correct; but ALT OUT DEST xxxx TRL NO HDR NO results in a syntax error.

Operand Definitions

Operand	Description								
DESTINATION <i>dest</i>	(Network printer only) Name of the destination printer.								
WIDTH <i>width</i>	(Network printer only) Width of the print line. For network printers, the value can be 80-250. Specify 0 to reset the width to the default value.								
HEADER <i>opt</i>	(Network printer only) Defines whether a header page prints at a network printer. For <i>opt</i> , specify YES to print a header page and NO to suppress printing a header page.								
TRAILER <i>opt</i>	(Network printer only) Defines whether a trailer page prints. For <i>opt</i> , specify YES to print a trailer page and NO to suppress printing a trailer page.								
FF <i>opt</i>	(Network printer only) Controls the PSS-generated formfeeds at the beginning and end of an output. <table border="1"> <tbody> <tr> <td>YES</td> <td>Issues formfeed at the start and end of print.</td> </tr> <tr> <td>NO</td> <td>Suppresses formfeeds.</td> </tr> <tr> <td>HEADER</td> <td>Issues formfeed at the start of print.</td> </tr> <tr> <td>TRAILER</td> <td>Issues formfeed at the end of print.</td> </tr> </tbody> </table>	YES	Issues formfeed at the start and end of print.	NO	Suppresses formfeeds.	HEADER	Issues formfeed at the start of print.	TRAILER	Issues formfeed at the end of print.
YES	Issues formfeed at the start and end of print.								
NO	Suppresses formfeeds.								
HEADER	Issues formfeed at the start of print.								
TRAILER	Issues formfeed at the end of print.								
PERTASK <i>num</i>	(Network printer only) Specifies the maximum number of outputs that can process during one print transaction for a specified printer. Note: For more information about FF <i>opt</i> and PERTASK <i>num</i> options, see Network Printer Definition Considerations.								
BLKSIZE <i>nnnnn</i>	(z/OS only) Specifies the maximum blocksize for output assigned to a data set when executing in batch. The default blocksize is 32 times the record length. Valid values are: <table border="1"> <tbody> <tr> <td>0</td> <td>The blocksize is the default--32 times the record length.</td> </tr> <tr> <td>1-65535</td> <td>The maximum blocksize, in bytes. The actual blocksize is calculated based on the record length, and equals the number of records that fit in the specified maximum blocksize.</td> </tr> </tbody> </table>	0	The blocksize is the default--32 times the record length.	1-65535	The maximum blocksize, in bytes. The actual blocksize is calculated based on the record length, and equals the number of records that fit in the specified maximum blocksize.				
0	The blocksize is the default--32 times the record length.								
1-65535	The maximum blocksize, in bytes. The actual blocksize is calculated based on the record length, and equals the number of records that fit in the specified maximum blocksize.								
DISPOSITION <i>disp</i>	Disposition for output to be printed at the named destination. The valid values are READY or HOLD.								

ALTER OUTPUT DESTINATION ALL DISPOSITION Command

The ALTER OUTPUT DESTINATION ALL DISPOSITION command changes the disposition for all output on ADROUT. The disposition of any new outputs created after execution of this command is not affected. Only existing outputs are changed.

This command has the following format:

```
ALTER OUTPUT DESTINATION ALL DISPOSITION disp
```

Operand Definitions

Operand	Description
DISPOSITION <i>disp</i>	Disposition for all output prints at the named destination. The valid values are: READY or HOLD.

ALTER PANEL RUN-STATUS Command

RUN-STATUS for panels and programs is an obsolete mechanism that has been replaced by Load Module format for production execution. This description is included here as an explanation of the attribute.

This command has the following format:

```
ALTER PANEL name VERSION nnn RUN-STATUS {PRIVATE }
                                         {RESIDENT}
```

Operand Definitions

Operand	Description
PRIVATE	(Default) Each user concurrently running the panel has a separate copy of the entire panel.
RESIDENT	Multiple users accessing the same panel share the portion of the panel that is not updateable.

ALTER PROGRAM DATAVIEW Command (CA Datacom CBS Access)

For a [assign the DAT variable value for your book] CBS dataview: The ALTER PROGRAM DATAVIEW command alters the dataview information in the object code for this program without the necessity of using the dictionary facility, making the change, recataloging the dataview, and then recompiling the program.

Note: Do not perform this command for a program that was the object of a CREATE MODULE. Perform the ALTER PROGRAM before the CREATE MODULE if you need to alter the dataview. For more information, see the *Administration Guide*.

This command has the following format:

```
ALTER PROGRAM name [VER version] DATAVIEW name DBID nnn
```

Operand Definitions

Operand	Description
VER version	Version commands display the version of the program accessing the dataview. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the "Preliminary Concepts" chapter.
DBID <i>nnn</i>	(For an CA Datacom CBS dataview) Specifies a different database ID.

ALTER PROGRAM DATAVIEW Command (VSE)

For a VSE unmodelled dataview: The ALTER PROGRAM DATAVIEW command alters the dataview information in the object code for this program without the necessity of using the dictionary facility, making the change, recataloging the dataview, and then recompiling the program.

You can issue this command for a program being developed in either an MVS or VSE environment. However, the block size or device type is only used when the program runs in VSE.

Note: Do not perform this command for a program that was the object of a CREATE MODULE. Perform the ALTER PROGRAM before the CREATE MODULE if you need to alter the dataview. For more information, see the *Administration Guide*.

This command has the following format:

```
ALTER PROGRAM name [VER version] DATAVIEW name
      [BLKSIZE number      ]
      [                    ]
      [      {DISK }      ]
      [      {SLTAPE}    ]
      [DEVICE {NLTAPE}   ]
      [      {PRT  }     ]
      [      {PUNCH }    ]
```

Operand Definitions

Operand	Description
VER version	Version commands display the version of the program accessing the dataview. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the “Preliminary Concepts” chapter.
BLKSIZE	(For a sequential file dataview) Specifies a new block size.
DEVICE	(For a sequential file dataview) Specifies a new device type.

ALTER PROGRAM DBID Command (CA Datacom CBS Access)

The ALTER PROGRAM DBID command changes all references to a DBID to another DBID. This applies to all dataviews in this program associated with the specified DBID. This command modifies the object code for the program without having to recompile the program. The DBID referenced is permanently altered in the object program until a subsequent ALTER or a recompilation takes place.

Note: Do not perform this command for a program that was the object of a CREATE MODULE. Perform the ALTER PROGRAM before the CREATE MODULE if you need to alter the database ID. For more information, see the *Administration Guide*.

This command has the following format:

```
ALTER PROGRAM name VERSION version DBID dbid-1 DBID dbid-2
```

Operand Definitions

Operand	Description
<i>version</i>	Version of the program. For more information about valid versions to specify, see the section <i>Version Clauses in CA Ideal Commands</i> in the chapter "Preliminary Concepts."
<i>dbid-1</i>	Old database.
<i>dbid-2</i>	New database to use.

ALTER PROGRAM ENVIRONMENT Command (CA Datacom SQL Access)

The ALTER PROGRAM ENVIRONMENT command changes the CA Datacom access plan options for the specified program in the object code for the program without having to recompile the program. The options changed are permanently altered in the object program until a subsequent ALTER or a recompilation takes place. To apply these changes to the run-time access plan, use the REBIND command. If you defined multiple plans for the program, you must rebind each plan where the altered plan options apply.

You can specify these options before compiling a program using the program environment fill-in (except the CLOSE option). You can specify default values for the compile using the SET DBSQL command (except the WORK and CLOSE options).

Note: Do not perform this command for a program that was the object of a CREATE MODULE. Perform the ALTER PROGRAM before the CREATE MODULE if you need to alter the plan options. For more information, see the *Administration Guide*.

This command has the following format:

```
ALTER PROGRAM name [VERSION version] [ENVIRONMENT option]
```

Operand Definitions

Operand	Description
<i>name</i>	One- to eight-character program name.
<i>version</i>	Version of the program. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the "Preliminary Concepts" chapter.
<i>option</i>	Access plan option. If you omit the ENVIRONMENT clause, a fill-in showing the current options from the program object displays. The options are shown in the table below:
Option	Description
CBSIO <i>nnnnnn</i>	I/O limit interrupt value for SQL statements that creates a set. nnnnnn a value from 0 to 524287.
CLOSE <i>time</i>	Specifies whether to close the plan after each database transaction or at the end of the run-unit or CICS job. TRAN close after each transaction. RUN close after the end of the run-unit.

Option	Description
DATE <i>date-format</i>	<p>Display format for SQL date type items.</p> <p>DB CA Datacom installed default; one of the following formats</p> <p>ISO International Standards organization: yyyy-mm-dd</p> <p>USA U.S. standard: mm/dd/yyyy</p> <p>EUR European standard: dd.mm.yyyy</p> <p>JIS Japanese Industrial Standard: yyyy-mm-dd</p>
DECPOINT <i>x</i>	<p>Character used as the decimal point when data displays. This has no effect on how data is stored.</p> <p>P Period (.) is used as the decimal point and comma is used as the digit separator.</p> <p>C Comma (,) is used as the decimal point and period is used as the digit separator.</p>
ISOLATION-LEVEL <i>level</i>	<p>Degree to which a unit of recovery is isolated from the updating operations of other units of recovery. Use for updates, deletes, or inserts.</p> <p>U No locks are acquired.</p> <p>R Repeatable read.</p> <p>CS Cursor stability.</p>
OPTMODE <i>optmode</i>	<p>How CA Datacom/DB Database (CA Datacom) optimizes the ordering of table joins.</p> <p>PREP Order joins during bind processing.</p> <p>MAN Order joins as specified in FROM clauses.</p> <p>EXEC Order joins at execution time. For more information, see the <i>CA Datacom/DB Database SQL User Guide</i>.</p>
PRI <i>nn</i>	<p>Priority level of the SQL requests.</p> <p>nn The lowest priority is 1, the highest priority is 15.</p>
STRDELIM <i>x</i>	<p>Character used to delimit string values in all SQL statements.</p> <p>A Apostrophes (') delimit string values.</p> <p>Q Quotation marks (") delimit string values.</p>
TIME <i>time-format</i>	<p>Display format for SQL time type items.</p>

Option	Description
DB	<p>CA Datacom installed default; one of the following formats:</p> <p>ISO International Standards Organization: hh.mm.ss</p> <p>USA U.S. standard: hh:mm AM or PM</p> <p>EUR European standard: hh.mm.ss</p> <p>JIS Japanese Industrial Standard: hh:mm:ss</p> <p>WAIT nnn unit Exclusive control wait limit.</p> <p>nnn 1 to 120 minutes or seconds.</p> <p>unit SEC for seconds or MIN for minutes. For example, for a ten-second wait, enter the command WAIT 10 SEC.</p>
Operand	Description
OPTMSGSGS time <i>type</i>	<p>Type of optimization messages CA Datacom/DB produces during bind processing or execution.</p> <p>time PREP to set the type of message produced during bind processing; or EXEC to set the type of message produced during execution.</p> <p>type</p> <p>NON None (default)</p> <p>DET Detail</p> <p>SUM Summary</p>
WORK space	<p>Amount of workspace available at plan execution time (can be increased in response to specific SQL errors).</p> <p>space 0 to 128 bytes, inclusive.</p>

ALTER PROGRAM RUN-STATUS Command

RUN-STATUS for panels and programs is an obsolete mechanism that has been replaced by Load Module format for production execution. This description is included here as an explanation of the attribute. The command is not applicable for programs in load module format. RUN-STATUS is not effective on VLS object, but the command is accepted to prevent syntax errors in existing procedures.

This command has the following format:

```
ALTER PROGRAM name [VERSION version] RUN-STATUS {PRIVATE }
                                                {SHARED }
                                                {RESIDENT}
```

All 3 options (PRIVATE, SHARED, RESIDENT) are treated as PRIVATE. Each user concurrently running the program has a separate copy of the entire object program.

ALTER SIGNON Command

The ALTER SIGNON command is used to change your password without editing your user definition or to change the greeting on the CA Ideal signon panel. This command has the following format:

```
ALTER SIGNON {PASSWORD old-password new-password}
              {PANEL }
              }
```

Operand Definitions

Operand	Description
PASSWORD	Changes the password required during signon. It allows you to change your own password without editing the user definition. You can use it on user definitions in production status. The new password specified here reflects in the user definition.
<i>old-password</i>	Current 1- to 12-character string assigned as the password.
<i>new-password</i>	New 1- to 12-character string replacing the old password.
PANEL	Changes the greeting message that appears on the CA Ideal sign-on panel. An image of the panel displays. You can type over the old message.

ASSIGN AUTHORIZATION Command (SQL Access)

The ASSIGN AUTHORIZATION (SQL Access) command overrides the authorization ID specified in the program resource table for a dataview for DB2 or in the environment fill-in for a program using CA Datacom SQL access.

For DB2 access: This command lets you change the authorization ID used to qualify tables or views accessed by a program without having to change the program definition. The command is issued before running a program that executes in dynamic mode and before generating the plan for a program that executes in static mode. (Issuing the command before running a program in static mode has no effect.)

For CA Datacom SQL access: This command is issued before running a program to select an alternate plan at runtime. As a result, the statements in the new plan are executed instead of the statements in the default plan.

This command has the following format:

```

                {UNQUALIFIED}    {UNQUALIFIED}
ASSIGN AUTHORIZATION {auth-id1 } NEW {auth-id2 }

```

Operand Definitions

Operand	Description
<i>auth-id1</i>	<p>For DB2: An authorization ID specified for a DB2 table or view in the program resource table.</p> <p>For CA Datacom SQL access: The authorization ID of the default plan entered in the environment fill-in of the program definition and stored in the program object.</p>
<i>auth-id2</i>	<p>For DB2: An authorization ID that overrides the specified authorization ID.</p> <p>For CA Datacom SQL access: The authorization ID of the alternate plan. This authorization ID must be defined before the ASSIGN command is entered. (See the DEFINE AUTHORIZATION command.)</p>
UNQUALIFIED	<p>(For DB2 only) As a modifier of old-auth-id, indicates that unqualified table references are to be qualified by new-auth-id.</p> <p>As a modifier of new-auth-id, indicates that table references qualified by old-auth-id are unqualified.</p> <p>UNQUALIFIED cannot be assigned to UNQUALIFIED.</p>

Example

Given the following program resource table definition:

DataviewAuth-id Qualify?	
EMPLOYEESBL	Y
PAYROLL HOU	N

The following embedded SQL statements change those generated FROM clauses as specified:

```
EXEC SQL      EXEC SQL
  SELECT ...  SELECT ...
    FROM EMPLOYEE, PAYROLL      FROM SBL.EMPLOYEE, HOU.EMPLOYEE
END-EXEC     END-EXEC

FROM SBL.EMPLOYEE, PAYROLL      FROM SBL.EMPLOYEES, HOU.PAYROLL
```

The following ASSIGN commands change those generated FROM clauses as specified:

```
No ASSIGN command entered
  FROM SBL.EMPLOYEE, PAYROLL      FROM SBL.EMPLOYEE, HOU.PAYROLL
ASSIGN AUTHORIZATION SBL NEW KRI
  FROM KRI.EMPLOYEE, PAYROLL      FROM KRI.EMPLOYEE, HOU.PAYROLL

ASSIGN ATZ HOU NEW KRI
  FROM SBL.EMPLOYEE, PAYROLL      FROM SBL.EMPLOYEE, KRI.PAYROLL

ASSIGN ATZ SBL NEW UNQUALIFIED
  FROM EMPLOYEE, PAYROLL          FROM EMPLOYEE, HOU.PAYROLL

ASSIGN ATZ UNQUALIFIED NEW KRI
  FROM SBL.EMPLOYEE, KRI.PAYROLL  FROM SBL.EMPLOYEE, KRI.PAYROLL
```

Note: This assignment affects CA Ideal programs only.

DB2 Considerations

DB2 qualifies unqualified names as follows:

- **For static mode:** Using the authorization ID of the user who performs the GENERATE PLAN.
- **For dynamic mode:** In batch, using the authorization ID of the user who runs the job; or online, using the auth-id based on the option for AUTH in the RCT entry for the current transaction.

Only a single level of authorization substitution is allowed. For example, if both ASSIGN ATZ SBL NEW HOU and ASSIGN ATZ HOU NEW KRI are issued, HOU replaces authorization ID SBL, not KRI.

If more than one assignment is made, only the first assignment applies. For example, if both ASSIGN ATZ SBL NEW HOU and ASSIGN ATZ SBL NEW KRI are issued, HOU replaces authorization ID SBL, not KRI. To reassign authorizations, use the RESET command and another ASSIGN command.

The ASSIGN AUTH command applies only to CA Ideal programs included in the plan. It does not affect any programs already bound into a package.

ASSIGN DATAVIEW Command (CA Datacom Native Access)

The ASSIGN DATAVIEW command associates the specified CA Datacom CBS dataview with a database ID for any applications run during the session. Use the RESET command to return to the session defaults.

The same command can also re-assign the table id for accessing a partitioned table, allowing access to an individual partition, or the ANY set.

This command has the following format:

```
ASSIGN DATAVIEW name [DBID nnn] [TABLE ttt]
```

Operand Definitions

Operand	Description
<i>name</i>	Name of the dataview to associate with the database.
<i>nnn</i>	Specifies the DBID temporarily selected for the dataview.
<i>ttt</i>	Specifies the table id to use for the dataview.

ASSIGN DATAVIEW Command (VSE Only)

It can also alter a sequential file dataview's block size, device type, or logical unit assignment for subsequent runs in the current session. This command has the following format:

```

                                [BLKSIZE number]
ASSIGN DATAVIEW name [DEVICE type ]
                                [TO SYSnnn   ]
    
```

Operand Definitions

Operand	Description
<i>name</i>	Name of the dataview to associate with the database.
BLKSIZE <i>number</i>	Temporarily alters the blocksize.
DEVICE <i>type</i>	Temporarily alters the device type where type is one of the following reserved words enclosed in quotes: DISK, SLTAPE, NLTAPE, PRT, or PUNCH.
TO <i>SYSnnn</i>	Temporarily alters the logical unit assignment for any subsequent runs in the current session.

ASSIGN DBID Command (CA Datacom Native Access)

The ASSIGN DBID command substitutes one database for another database during a session. Use the RESET command to return to the session default.

This command has the following format:

```
ASSIGN DBID dbid-1 NEW dbid-2
```

Operand Definitions

Operand	Description
<i>dbid-1</i>	Old database.
<i>dbid-2</i>	New database to use.

ASSIGN PROGRAM Command

The ASSIGN PROGRAM command substitutes a test version of a CA Ideal subprogram for the production or another test-status version of that program during a run. This lets you test a subprogram with the production version application without replacing the production version subprogram or creating a test version of the entire application.

This command only affects applications for the current CA Ideal session. Use the RESET command to return to the session default.

This command has the following format:

```
ASSIGN PROGRAM name [SYS sid] [T0] [NEW] {VER nnn}
                               {SYS sys}
```

Operand Definitions

Operand	Description
	A CA Ideal program that currently has a production version and at least one test version. Specifying a non-ideal subprogram results in the ASSIGN command being ignored.
<i>sid</i>	Three-character ID of the system containing the production (or old test) version of the program. This is required only if the old program is in a system other than the current system.
<i>nnn</i>	Version number of the program being tested. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the "Preliminary Concepts" chapter.
<i>sys</i>	Three-character ID of the system containing the new test version of the program. This is required only if the new program is in a system other than the current system or, if <i>sid</i> is specified, if the new program is not in the system <i>sid</i> .

ASSIGN REPORT Command

The ASSIGN REPORT command specifies report output parameters for the duration of the current session when issued before a run. During one run, the maximum number of active ASSIGN REPORT commands is the number of reports being generated, to a maximum of 20 per user.

The parameters specified in the ASSIGN REPORT command override the corresponding parameters specified in the RUN command. Parameters specified in a subsequent ASSIGN REPORT command override the corresponding parameters in the previous ASSIGN REPORT command. Use the RESET command to return to the parameters specified in the RUN command or to the session defaults.

You must specify at least one of the options in the command.

This command has the following format:

```
ASSIGN REPORT name [TO altname]
[      {MAIL 'email-id'          } ]
[DESTINATION {LIBRARY           } ]
[      {{SYSTEM name           } } ]
[      {{NETWORK 'name'} [COPIES n]} ]
[                                         ]
[DISPOSITION 'disp'                ]
[MAXLINES m                          ]
[DESCRIPTION 'string'               ]
[PAGE SIZE nnn                       ]
```

Operand Definitions

Operand	Description
<i>name</i>	Name of the report directed by the ASSIGN REPORT command or the word RUNLIST. Specifying RUNLIST determines the destination of the LIST statement output.
<i>altname</i>	Alternate name for the report, as a ddname (in z/OS) or as SYSnnn (in VSE). (Not available under CICS.)
DESTINATION	<p>MAIL 'email-id' A delimited 1- to 60-character name of a [assign the value for emailp in your book] destination enclosed in single or double quotes.</p> <p>LIBRARY Output library.</p> <p>SYSTEM name System printer name.</p> <p>NETWORK name Network printer name (not available in batch).</p>

Operand	Description
COPIES <i>n</i>	Specifies the number of copies for a destination of system or network in an online environment only. (This clause is ignored in batch.) This option is only valid when specifying DESTINATION NETWORK or DESTINATION SYSTEM.
DISPOSITION ' <i>disp</i> '	Keywords include: 'KEEP' 'RELEASE' 'HOLD' For more information about output dispositions, refer to the section titled SET OUTPUT command.
MAXLINES <i>m</i>	Maximum number of lines for any one report that a program can produce. The upper limit for MAXLINES is established at CA Ideal installation or by a SET OUTPUT SITE OPTIONS fill-in. It only applies to reports going to the output library. Any report reaching this maximum stops the run. This entry does not affect reports produced in batch and printed in batch.
DESCRIPTION ' <i>string</i> '	One- to 32-character description.
PAGE SIZE <i>nnn</i>	Maximum number of lines, ranging from 1 through 250, for each page of the report. This overrides the number of lines per page set in the report parameter section.

BLANKFILL Command

The BLANKFILL command is available only in the Panel Definition Facility (PDF). It specifies the panel layout fill mode. To edit the panel layout without padding, see the NULLFILL command.

Note: PF21 toggles between blankfill and nullfill mode.

This command has the following format:

```
BLANKFILL
```

CATALOG DATAVIEW Command

The CATALOG DATAVIEW command or equivalent CATALOG prompter catalogs a dataview using the CATALOG command. You can also access the CATALOG prompter by selecting Option 5 on the Administration Maintenance Menu or Option 2 on the Dataview Menu. One or many CATALOG commands can be issued in a batch run of CA Ideal.

A dataview must be cataloged before CA Ideal can use it. When the CATALOG command is processed, a DISPLAY DATAVIEW command is automatically invoked. CATALOG results in a cataloged dataview definition ready for CA Ideal use. If the dataview is in production status, any CA Ideal program naming the dataview as a resource that compiles successfully can execute.

If one or more rules are violated, a list of error messages displays at the bottom of the dataview display (or, if the CATALOG command is issued using CA Ideal in batch, a listing is printed).

This command has the following format:

```
          {          *          }
CATALOG {          {authid.dvw-name } }
          {[dbms] DATAVIEW {dvw-name [VERSION ver]} }

          [          {VALIDATION } ]
          [[WITH] {NOVALIDATION} ]
```

Operand Definitions

Operand	Description
*	An asterisk represents the current dataview. You can substitute it for the entire DATAVIEW identification clause.
<i>dbms</i>	<p>(Mixed SQL Site) Database management system that SQL statements using this dataview access when you can access more than one database management system using SQL. This option overrides the <i>dbms</i> specified by a SET ENVIRONMENT SQL command. Use this option when you catalog a new SQL dataview.</p> <p>To change the database management system for a cataloged dataview, delete the dataview and catalog again. You cannot make the change when recataloging an existing SQL dataview.</p> <p>DB Dataview accesses CA Datacom/DB. DB2 Dataview accesses DB2.</p>
<i>authid</i>	(SQL Only) One- to eight-character authorization ID required for SQL dataviews only.
<i>dvw-name</i>	The 1- to 18-character name of a dataview. For SQL dataviews, the dataview name is the name of an SQL object (table, view, or CA Datacom SQL synonym).
<i>ver</i>	(Non-SQL dataviews) Version of the dataview to catalog, for a modeled (CA Datacom native access or sequential) dataview or an unmodeled (sequential or VSAM) dataview. If the version is not specified, the default version set by the SET DATAVIEW VERSION command is used. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the "Preliminary Concepts" chapter.

Operand	Description
[WITH] VALIDATION	<p>(VSAM Dataviews Only) Specify this option to verify CA Ideal dataview attributes against the actual characteristics of the VSAM file in the VSAM Access Control Blocks. If any discrepancies are found, the CATALOG command fails and error messages are issued. You can specify this option as VAL.</p> <p>The following attributes are validated:</p> <ul style="list-style-type: none">■ File organization (ESDS, KSDS, RRDS)■ Record length (maximum length must match the length defined in IDCAMS)■ Displacement and length of primary and alternate keys <p>UNIQUE and UPGRADE SET attributes of the alternate indexes are not validated, although paths defined as part of the upgrade set are checked for availability and the key defined for each path is checked for displacement and length.</p> <p>Use this option only if the file is accessible in the current environment.</p>
[WITH] NOVALIDATION	<p>(VSAM Dataviews Only) Specify this option to catalog the dataview without validating the attributes. You can specify this option as NOVAL.</p> <p>WITH NOVALIDATION is the default, but you can change the default by specifying the command SET CATALOG VALIDATION.</p> <p>Use this option if the file is not accessible in the current environment.</p>

CLARIFY Command

The CLARIFY command displays a panel with messages explaining violations of editing rules. The CLARIFY command is only available if it is specified in the panel parameter fill-in.

When an entry in a field of a panel violates the rules defined for the field, the panel redisplay with the fields in error set as specified in error handling (see the parameter fill-in and extended field definition fill-in descriptions). These edit rule violations are detected and highlighted when a panel is transmitted at runtime or when the panel is tested with the FACSIMILE command during panel definition.

Violations of panel input edit rules can also be handled by the application procedure instead of by the CLARIFY FACILITY. You can also specify this option on the parameter fill-in. The application procedure can make use of PDL internal functions to determine the rule violations. Messages to explain the edit rule violations are listed on a CLARIFY panel that you can display by pressing the PF3 key or the entering the CLARIFY command.

You can scroll the CLARIFY panel forward or backward and left or right. The RETURN command, PF2, returns from the CLARIFY panel to the original panel.

This command has the following format:

```
CLARIFY
```

COMBINE Command

The COMBINE command eliminates the last region when in split screen. This combines regions from the bottom up. If there are three regions, the second and third regions combine restoring region 2. Region 2 now occupies the space occupied by regions 2 and 3 before the COMBINE. (The size of region 1 remains unchanged.) If you issue another COMBINE command, the first and second regions combine, leaving the first region restored.

Each step requires a COMBINE command; that is, to combine three regions, specify two separate COMBINE commands. The OFF command performs the same function.

This command has the following format:

```
COMBINE
```

Comment

A comment is a character string that serves as documentation for a CA Ideal command in a member or in a batch jobstream. Comments are not executable. A line that begins with a colon is treated as a comment. You can place a comment character before any command to prevent its execution. Because the comment ends at the end of the line, no special character is required to terminate the comment.

A comment is indicated by a character set for the site or session by the SET COMMAND COMMENT command or the installed default-a colon.

Example

Batch Jobstream

```
DUP PGM A PROD NEXT      : copy prod ver
COMPILE *                 : compile it
MARK * PROD              : make it prod
```

COMPILE Command

The COMPILE command compiles a TEST status program in the current system. This command is executed asynchronously where you can continue with other tasks while the COMPILE command is executing. COMPILE alone produces the COMPILE prompter. COMPILE * compiles the current program.

You can only compile test programs in the current system. If you enter the COMPILE command for a production program (PROD status), the program does not recompile, but a compilation listing is generated.

The working data, parameter data, and dataview panels are only compiled if the corresponding source was changed since it was previously compiled.

This command has the following format:

```
[* ]
COMPILE [program-name [VERSION version]]

[ {YES}] [ {YES}] [ {YES}]
[IDE {NO }] [EXD {NO }] [BOD {NO }]

[ {YES}] [ {YES}] [ {YES}]
[ADV {NO }] [MEL {NO }] [LSQL {NO }]

[ {FULL}] [ {FULL}]
[REF {SHORT}] [PANEL {SHORT}]
[ {NO }] [ {NO }]

[ {MAIL 'email-id' } ]
[ {LIBRARY } ]
[DESTINATION {{NETWORK name} } ]
[ {{SYSTEM name } [COPIES nn] } ]

[NAME output-name]

[ {KEEP } ]
[DISPOSITION {RELEASE}]
[ {HOLD } ]

[DESCRIPTION 'string']
```

Operand Definitions

Operand	Description
<i>program-name</i>	One- to eight-character name of the program to compile.
<i>version</i>	Version of the program. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the “Preliminary Concepts” chapter.
<i>IDE opt</i>	Y Includes program identification, resources, and environment sections in the compilation listing. N Does not include identification information in the compilation listing.
<i>EXD opt</i>	Y Includes external data (dataview definitions, report definitions, panel definitions, and copied SQLCAs) in the compilation listing. N Does not include external data in the compilation listing.

Operand	Description
<i>BOD opt</i>	<p>Y Includes the body of the program (working data, parameter data, and procedure definition) in the compilation listing.</p> <p>N Does not include the body of the program in the compilation listing.</p>
<i>ADV opt</i>	<p>Y Includes advisory messages in the compilation listing.</p> <p>N Does not include advisory messages in the compilation listing.</p>
<i>MEL opt</i>	<p>Y Highlights error lines in the original procedure definition.</p> <p>N Does not highlight error lines.</p>
<i>LSQL opt</i>	<p>Y Lists SQL generated from any FOR constructs.</p> <p>N Does not list generated SQL.</p>
<i>REF opt</i>	<p>FULL Generates a complete cross reference listing.</p> <p>SHORT Suppresses those symbol names that are defined but not referenced.</p> <p>NO Does not generate any cross reference information.</p> <p>You can specify this option for online compiles and for any batch compiles.</p>
Program-name	One- to eight-character name of the program to compile.
<i>PANEL opt</i>	<p>FULL Generates a complete listing of all panel components.</p> <p>SHORT Generates only the Identification, Facsimile, and Summary associated with each panel.</p> <p>NO Does not generate any panel listings. This option applies only if EXD is Y.</p>
DESTINATION	<p>MAIL 'email-id' Delimited 1- to 60-character name of a [assign the value for emailp in your book] destination.</p> <p>LIBRARY Output library.</p> <p>SYSTEM name System printer name.</p> <p>NETWORK name Network printer name (not available in batch).</p>
<i>COPIES nn</i>	<p>Number of copies to print on a system or network printer. This option is ignored in batch.</p> <p>Note: When DESTINATION is the output library, only one copy is allowed.</p>
<i>NAME output-name</i>	Name of the output. The default is the eight-character name of the program if compiled online or COMPLIST if compiled in batch.
DISPOSITION	<p>KEEP</p> <p>RELEASE</p> <p>HOLD</p>

Operand	Description
DESCRIPTION 'string'	The 1- to 32-character description.

COPY Command

The COPY command copies all or part of a program's working data, parameter data, procedure, or all or part of a member. If neither a program nor member name is specified, data is copied from one specified destination in the current program or member to another specified destination in the current program or member.

In the Panel Definition Facility (PDF), the COPY command invokes the panel copy prompter. The panel copy prompter copies dataview field definitions into the panel layout. For more information, see the *Creating Panel Definitions Guide*.

This command has the following format:

```

      [          [  {nnn } ]          ]
      [PROGRAM name [VER {PROD}] [SYS id]]
      [          [  {LAST} ]          ]
COPY [          [          ] [num-1 [num-2 ] ]
      [          [          ] [TOP  [BOTTOM ] ]
      [MEMBER name [USER {name}] ]
      [          [  {id } ]          ]

      {num-3 }
      {TOP  }
      {BOTTOM}

```

Operand Definitions

Operand	Description
PROGRAM <i>name</i>	One- to eight-character name of the program definition from which data is copied. The component from which data is copied corresponds to the component being edited in the current program.
VER <i>nnn</i>	Version of the program definition from which data is copied. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the "Preliminary Concepts" chapter.
SYS <i>id</i>	Name of the system where the program resides. This is required when copying from a different system.

Operand	Description
MEMBER <i>name</i>	One- to eight-character name of the member from which data is copied.
USER <i>clause</i>	User name or the one- to three-character user ID where the member resides. This is required when copying from a different user.
<i>num-1</i> TOP	Specification for the beginning of a range of lines to be copied. If neither a program name nor a member name is specified, data is copied from the specified location in the current program or member to the specified location in the same program or member. num-1 The sequence number of the line to copy or, if specified with num-2 or BOTTOM, the first line of a range of lines to copy.
TOP	Indicates the first line in the program definition or member.
<i>num-2</i> BOTTOM	Specification for the ending of a range of lines to copy. num-2 The sequence number of the line marking the end of the range to copy. BOTTOM Indicates the last line in the program or member
<i>num-3</i> TOP BOTTOM	Specification for the COPY destination. num-3 The sequence number of the line in the component of the current program definition or the current member after which the line or lines being copied are inserted. TOP Indicates the top of the component of the current program definition or member. BOTTOM Indicates the bottom of the component of the current program definition or member.

CREATE DATAVIEW Command

The CREATE DATAVIEW command initiates creation of a dataview for a specific unmodeled dataview (sequential or VSAM). For more information about defining unmodeled sequential or VSAM dataview, see the *Creating Dataviews Guide*.

This command has the following format:

```
CREATE [DATAVIEW dvw-name]
```

Operand Definitions

Operand	Description
program-name	One- to eight-character name of the program to compile.
<i>dvw-name</i>	The 1- to 18-character dataview name of a sequential or VSAM file dataview.

CREATE MEMBER Command

The CREATE MEMBER command initiates creation of a member.

This command has the following format:

```
CREATE MEMBER name [USER user-name] [DESCRIPTION 'string']
```

Operand Definitions

Operand	Description
<i>name</i>	One- to eight-character member name.
<i>user-name</i>	The 1- to 15-character user name or the one- to three-character user-id. The user-name is required only if the member belongs to another user.
<i>'string'</i>	The 1- to 24-character description of the member being created.

Example

```
CREATE MEMBER JCLDEMO
CREATE MEMBER JCLDEMO DESCRIPTION 'DEMONSTRATION'
```

CREATE MODULE Command

The CREATE MODULE command creates the module form of a program or panel. This *batch-only* command is used the first time a program or panel is converted to module format. Use it also when a new PROD version of the same program or panel is converted to module format. As long as the same one- to seven-character module name is used, the CREATE MODULE command replaces the old module. However, if the module name for the program or panel changes, first delete the old module with the DELETE MODULE command. For more information, see the *Administration Guide*.

The primary output of a CREATE MODULE command is a set of IBM-format object decks, along with appropriate Linkage Editor control statements, written to a sequential file (three object decks per CA Ideal program and 1 object deck per panel). This file is used as input to the IBM Linkage Editor. Regardless of the number of VLS object entities, there is always one object deck module created for each of the reentrant, non-reentrant (updateable), and symbol table portions of each CA Ideal program.

You can specify many CREATE MODULE commands in the same CA Ideal batch run. The resultant object decks and appropriate Linkage Editor control statements are concatenated into the same sequential output data set. You must specify a job step following the CA Ideal batch step to execute the Linkage Editor and read that sequential file as its input.

Execute a SELECT SYSTEM command before the CREATE MODULE command to identify the CA Ideal system that contains the program. Do not execute RUN commands in the same CA Ideal job step as CREATE MODULE commands or the results can be unpredictable.

This command has the following format:

```
                                {PANEL name VERSION nnn}  
CREATE MODULE modname [FROM] {PROGRAM name      }
```

Operand Definitions

Operand	Description
<i>modname</i>	One- to seven-character name given to the generated modules.
<i>name</i>	Name of the program or panel to convert to module format. The program or panel must be in production status to be converted. If there are multiple versions, the one that is in production status is the one that is converted.

Note: For PANELs, if you omit the VERSION clause, the PROD version is used if it can be found by accessing Datadictionary. For PANEL MODULEs created from VLS objects that were Object Transported, the VERSION clause is necessary, because the Object Transport did not update Datadictionary. It is advisable to always specify the VERSION clause for panels.

The names of the created modules are the one- to seven-character modname with an one-character suffix. The suffixes are:

- **R** Reentrant (non-updateable) part of a program
- **U** Updateable (non-reentrant) part of a program
- **S** Symbol table of a program
- **P** Panel

These suffixes are used in the following cases:

- The user wants to create CICS PPT entries. This is not necessary unless the modules are CICS resident. Program auto-install defines them as CICS non-resident. For more information, see the *Administration Guide*.
- The user wants to use a standard IBM utility to move, delete, and so on, the modules in the system libraries.
- The user wants to obtain performance monitor data about the use of modules in the system, for example, CICS shutdown statistics that show how many times a module is used.

When dealing with non-Ideal services, the user must account for all three program modules and must specify the appropriate module suffix for each of these parts, and for any panel modules.

Important! Do not perform an ALTER PROGRAM command in a program that was the object of a CREATE MODULE. Perform the ALTER PROGRAM before the CREATE MODULE if you need to alter the database ID.

CREATE PACKAGE Command (DB2 Only)

The CREATE command or equivalent CREATE prompter creates a new package definition for an application accessing a DB2 database. This command presents a blank identification fill-in. You can edit this fill-in using standard editing procedures. The DB2 application package is not actually created until the successful completion of the batch job described in *Generating Application Packages in the Administration Guide*.

To display the CREATE prompter, select Option 2 on the Plan Menu or type the CREATE command. See also CREATE PLAN.

This command has the following format:

```
CREATE PACKAGE [name]
```

Operand Definitions

Operand	Description
<i>name</i>	One- to seven-character name assigned to the package definition. The name must consist only of legal characters for DB2 package names because it is the name of the DB2 application package. The identification fill-in and the other components of the package definition-resources and parameters are described in the <i>Administration Guide</i> .

CREATE PANEL Command

The CREATE PANEL command or equivalent prompter creates the first version of a panel definition. The only function of the CREATE PANEL command or prompter is to present a blank panel identification fill-in. This fill-in establishes the panel name, provides identification information, and becomes the current definition.

CA Ideal assigns a newly created panel a version number of 1. This version of the panel definition is in TEST status. You can edit it at any time as long as it remains in TEST status. For more information about how to edit the procedure, see the *Creating Panel Definitions Guide*.

This command has the following format:

```
[ [[USING] [* ]]]  
CREATE PANEL [name [[FROM ] [Dwv dvw-name VERSION ver]]]
```

Operand Definitions

Operand	Description
<i>name</i>	One- to eight-character name assigned to the new panel.
USING FROM	Creates a panel containing the fields defined in the specified dataview.
*	Creates a panel using the current dataview.
<i>dvw-name</i>	The 1- to 32-character name of a cataloged dataview from which the fields are copied. An CA Datacom native access dataview name has a maximum of 18 characters. An SQL dataview name has a maximum of 27 characters and must be qualified with the authorization ID. A sequential dataview name has a maximum of 32 characters.
<i>ver</i>	Version of the named dataview specified as PROD, <i>nnn</i> , or for modeled dataviews, <i>Tnnn</i> . The default is PROD. The keyword VERSION is optional when specifying the version as PROD

Using Dataview with the CREATE Command

When a dataview is named on the CREATE command, the dataview fields are copied into the panel after you enter the identification fill-in. All fields and values that can be reasonably defined in the panel definition are copied. The following rules apply:

Data Type

- Any alphanumeric field is copied as is. Since a fixed-length value is required for the panel, the maximum length of a variable length dataview field is assumed to be the panel field length. If the length of a dataview field exceeds the maximum width of a panel, the field is truncated to the maximum width less 1 (max-width - 1). This applies to all copied fields, whether they are copied on the CREATE command or with the copy prompter.
- Any numeric field is copied as an unsigned numeric field, regardless of the internal definition. Signed and unsigned numeric fields in the dataview are copied as numeric fields without regard to sign.
- The number of integer and decimal places denoted in the panel definition as IN.DP is copied directly from the dataview field definition. The size of the panel field is based on the total number of integer and decimal places plus one for a decimal point if so defined. The total for integer and decimal places cannot exceed 18.
- Any CA Ideal date field is copied as a numeric field. Once defined in the panel, you can change the data type to alphanumeric, depending on the date display format.
- Any condition-name is ignored.

Groups

- Any field that is or can be reduced to an elementary field is copied. REDEF fields and group fields are copied as elementary fields, but fields with DEP ON are not copied.
- Nesting of OCCURS groups is ignored.

Field Attributes

- Initial values, if specified, are copied.
- Edit patterns, if specified, are copied. However, the size of the panel field is taken from the defined length of the dataview field and does not necessarily account for the length of the edit pattern. You might need to modify the panel field after it is copied from the dataview.
- The panel field name is the same as the dataview field name. The dataview qualification for panel field names is dropped since CA Ideal allows only one level of group qualification.
- The panel field label is generated from the cataloged dataview heading, if defined, or from the panel field name, if a cataloged dataview heading is not defined.

CREATE PLAN Command (DB2 Only)

The CREATE PLAN command presents a blank identification fill-in. You can edit this fill-in using standard editing procedures. The DB2 application plan is not actually created until the successful completion of the batch job described in Generating Application Plans in the *Administration Guide*. The identification fill-in and the other components of the plan definition-resources, parameters, and DBRMs-are further described also.

The following CREATE command or equivalent CREATE prompter creates a new application plan definition for an application accessing a DB2 database. To display the CREATE prompter, select Option 2 on the Plan Menu or type the CREATE command.

See also CREATE PACKAGE.

This command has the following format:

```
CREATE PLAN [name]
```

Operand Definitions

Operand	Description
<i>name</i>	One- to seven-character name assigned to the plan definition. The name must consist only of legal characters for DB2 plan names because it is the name of the DB2 application plan.

CREATE PROGRAM Command

The CREATE PROGRAM command or equivalent CREATE prompter creates the first version of a new program definition. The sole function of the CREATE PROGRAM command or prompter is to present a blank program identification fill-in (see Creating Programs). This fill-in establishes the program name and provides identification information for the program.

CA Ideal assigns a newly created program definition a version number of 1. This version of the definition is in TEST status. You can edit it at any time as long as it remains in TEST status. After it is marked to PROD (production) status, you cannot modify it.

After the CREATE command is executed, the identification fill-in is presented, and you enter a program name, that program definition becomes the current definition, and the editing procedure described in the *Creating Programs Guide* applies.

To access the CREATE prompter, select Option 2 on the Program Maintenance Menu or type CREATE PROGRAM on the command line.

This command has the following format:

```
CREATE [PROGRAM [name]]
```

Operand Definitions

Operand	Description
<i>name</i>	One- to eight-character name assigned to the new program definition.

CREATE REPORT Command

The CREATE REPORT command or prompter creates the first version of a report definition by displaying a blank report identification fill-in. This fill-in establishes the report name and provides identification information. When you enter the report name, it becomes the current report and the editing procedure applies.

CA Ideal assigns a newly created report definition a version number of 1. This version of the report definition is in TEST status. You can edit it at any time as long as it remains in TEST status. After it is in PROD (production) status, you cannot modify it.

To display the CREATE prompter, select Option 2 on the Report Maintenance Menu or type CREATE on the command line and press the Enter key.

This command has the following format:

```
CREATE REPORT [ {USING} {* } ]
              [name {FROM } DW {auth-id.dvw-name } ]
              {dvw-name [VERSION ver]}
```

Operand Definitions

Operand	Description
<i>name</i>	One- to eight-character name assigned to the new report definition.
<i>auth-id</i>	One- to eight-character authorization ID required for SQL dataviews.
USING FROM	Creates a report containing the fields defined in the specified dataview.
*	Creates a report using the current dataview.
<i>dvw-name</i>	The 1- to 18-character dataview name.
<i>ver</i>	(Non-SQL dataviews) Version of the dataview, for an CA Datacom native access dataview, a sequential file dataview, or a VSAM file dataview. The specified version must be in test or production status. For more information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the chapter "Preliminary Concepts."

CREATE SYSTEM Command

The CREATE command or equivalent CREATE prompter creates the first version of a system definition. The sole function of the CREATE SYSTEM command or prompter is to present a blank system definition fill-in. This fill-in establishes the system name, provides identification information about the system, and provides the file names where the system libraries are stored. Once the system definition fill-in is presented and you enter a system name, that system becomes the current system definition.

A newly created system definition is assigned a version of 1 and is placed in test status. You can edit the new system definition at any subsequent session as long as it remains in test status. Once marked to production status, you cannot modify it. For additional information, see the *Administration Guide*.

To access the CREATE prompter, select Option 2 on the System Maintenance Menu.

This command has the following format:

```
CREATE [SYSTEM {name}]
```

Operand Definitions

Operand	Description
<i>name</i>	The 1- to 15-character name assigned to the new system definition. The system name in the system definition fill-in is initialized to this entry. If omitted, you must enter the name as the system name on the system definition fill-in.

CREATE USER Command

The CREATE USER command creates the first version of a user definition. A newly created user definition is assigned a version number of 1. This version of the user definition is in test status. Before this user can sign on to CA Ideal, the user definition must be marked to production status (see the MARK STATUS command). You can modify it at any subsequent session as long as it remains in test status.

For more information on creating and maintaining user definitions, see the *Administration Guide*.

See also CREATE SYSTEM.

This command has the following format:

```
CREATE USER [name]
```

Operand Definitions

Operand	Description
<i>name</i>	The 1- to 15-character name assigned to the new user definition. The person's name on the user definition fill-in is initialized to this entry. If omitted in the CREATE command, you must enter the name as the person's name on the user definition fill-in. The user definition fill-in is a panel establishes the user, assigns a user privilege, enters descriptive information about the user, and establishes the user's authorization level in each assigned system.

DATAQUERY Command

The DATAQUERY command transfers directly to CA Dataquery from CA Ideal. DATAQUERY, DQRY, and DQ are valid synonyms.

This command has the following format:

```

        [USER user-name      ]
        [PASSWORD password  ]
        [                    ]
DATAQUERY [RETURN {NONE  }  ]
        [          {tran-id} ]
        [                    ]
        [COMMAND 'command string' ]
    
```

Operand Definitions

Operand	Description
<i>user-name</i>	CA Dataquery user-ID. You can use delimiters around the user-ID if it contains spaces. Otherwise, delimiters are not required. If you omit the USER clause, the program passes the CA Ideal user-ID.
<i>password</i>	CA Dataquery password. If you omit the password, the program passes the CA Ideal password.

Operand	Description
RETURN	Specifies a CICS transaction ID scheduled by CA Dataquery when it returns to CA Ideal. tran-id A CICS transaction ID scheduled by CA Dataquery. NONE The program supplies spaces as the return transaction-ID. If you omit the RETURN clause, CA Dataquery returns to CA Ideal with the transaction-ID used to sign on to CA Ideal.
' <i>command string</i> '	First CA Dataquery command executed. The commands must be in delimiters. If you do not supply a COMMAND string, the program passes spaces.

Example

```
DATAQUERY COMMAND /LIST QUERIES/
```

DATAVIEW Command

The DATAVIEW command illustrates the functions CA Ideal offers to manipulate dataviews in the Dataview Menu. To access this menu, select the appropriate option on the main menu or enter the following CA Ideal command.

This command has the following format:

```
DATAVIEW
```

DEBUG Command

The DEBUG command initiates the debugging of a program. When the DEBUG command is issued, an Initialization breakpoint appears on the screen that lets you specify (or alter) Debug commands. When a breakpoint is encountered, application processing stops, any commands attached to the breakpoint are executed, and the Debug screen displays at the terminal. When the Debug screen is transmitted, a checkpoint is issued that commits records marked for update. The SET RUN UPDATE NO command stops the application from updating the database. For more information on breakpoints, see the *Programming Reference Guide*. For more information about using the symbolic debugger, see the *Creating Programs Guide*.

Online: If no member name is specified, member DEBUG is used. If member DEBUG already exists, the contents of the member are saved.

Batch: Follow the DEBUG command with debug commands for the session and end with a GO command. When the GO command is encountered, execution of the program begins. When a breakpoint is encountered, any attached commands execute and application processing resumes. Output is written to the Debug print file. If no member name is specified, a member unique to this run is used.

Both online and batch: If a member name is specified, it is created if it does not already exist. If this member already exists, the contents of the member are saved.

Issuing the DEBUG command alone invokes the DEBUG prompter. DEBUG * debugs the current program.

This command has the following format:

```

DEBUG      { *                               }
           {pgm-name [VERSION version]}

           {Y}
           [UPDATE [DB] {N}                  ]
           [PARAMETER 'string'               ]
           [                                  ]
           [          {MAIL 'email-id'       }]
           [          {LIB                   }]
           [DESTINATION {{SYS name}          }]
           [          {{NET name} [COPIES num]}]
           [                                  ]
           [          {KEEP   }              ]
           [DISPOSITION {RELEASE}            ]
           [          {HOLD   }              ]
           [                                  ]
           [MAXLINES n                       ]
           [                                  ]
           [DESCRIPTION 'string'             ]
           [                                  ]
           [COMMANDS [{*                      }] ]
           [          [{member [USER uuu]}]   ]
    
```

Operand Definitions

Operand	Description
*	Debugs the current program.
<i>pgm-name</i>	Name of the program being debugged.

Operand	Description
<i>version</i>	Version of the program being debugged. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the “Preliminary Concepts” chapter.
UPDATE	Either updates or bypasses updating the database. Y Allows updating of the database. N Does not allow updating of the database.
PARAMETER 'string'	A single string passed to the program. This parameter must be defined in the parameter definition fill-in for the program being debugged as one alphanumeric (type X) character string with input Update Intent. It cannot be a group item. The length of the string is limited to the space available in the prompter. When the DEBUG command is issued in the command area, the command with parameter string is limited to one command line.
DESTINATION	Applies to all reports generated by this program unless this destination information was overridden for one or more reports with the ASSIGN REPORT command. This clause differs from the more typical DESTINATION-clause in that “NAME name” is excluded. The one- to eight-character report name becomes the output name. Select from the following: MAIL 'email-id' A delimited 1- to 60-character name of a [assign the value for emailp in your book] destination. LIBRARY Output library. SYSTEM name System printer name. NETWORK name Network printer name (not available in batch).
COPIES <i>num</i>	Number of copies to print by a system or network printer in an online environment. Cannot exceed site maximum. This clause is ignored in batch.
DISPOSITION	The disposition clause; select from the following: KEEP RELEASE HOLD For more information on output dispositions, see the SET OUTPUT Command.

Operand	Description
MAXLINES <i>n</i>	Maximum number of lines for any one report that a program can produce. The upper limit for MAXLINES is established at CA Ideal installation or by a SET OUTPUT SITE OPTIONS fill-in, and only applies to reports going to the output library. Any report reaching this maximum stops the run. This entry does not affect reports produced in batch and printed in batch.
DESCRIPTION ' <i>string</i>	The 1- to 32-character description of the output. The description displays as part of the output status. The default description DEBUGGER OUT applies if you do not provide a description in the DEBUG command. For more information about the output status description, see the <i>Working in the Environment Guide</i> .
COMMANDS	Specifies the name of the data member used for debugging commands. * Current member from the status line. member Name of the data member used for the debugging commands. The default is DEBUG. uuu User-ID of the data member. The default is the current user.

DEFINE AUTHORIZATION Command (CA Datacom SQL Access)

The DEFINE AUTHORIZATION command creates an alternate plan for a program that uses CA Datacom SQL access. The new plan is identified by a name created from the authorization ID, program name, and version number specified in the DEFINE AUTHORIZATION command. You can enter this command as DEFINE AUTHORIZATION or as DEFINE ATZ, DEFINE AUT, or as DEFINE AUTH. You must enter the command from the system that contains the program object. It cannot create the plan from a program in load module format. This command *must* be executed before a load module for the program is created in the environment. This system name is also used in the new plan name. If a plan already exists with the same name (including the authorization ID), that plan is deleted before the new plan is created. The new plan then takes the place of the old one.

The only thing that changes for the new plan is the authorization ID. The SQL statements are extracted from the existing program object. In the new plan, statements that reference resources defined in the program resource fill-in as QUAL=N are referenced with the authorization ID specified in the DEFINE AUTHORIZATION command. All other references in the new plan remain the same as in the original plan.

The DEFINE AUTHORIZATION command can also be used to create a plan after a CA Ideal program has been object transported to a production environment. Use of this command replaces the need to transport the plan from the development environment using the utility DDTRSLM.

The statement-IDs for each statement must match in every plan associated with the program or a runtime error occurs. This type of error can happen if the CA Datacom utility DDTRSLM exports a plan without its associated program, or if the CA Ideal Object Transport Utility transports a program object without its associated plan.

If you need to change plan options for the new plan, use the ALTER PROGRAM ENVIRONMENT command to change the program object after creating the new plan. Then enter the REBIND command, specifying the authorization ID of the new plan on the REBIND command.

This command has the following format:

```
DEFINE AUTHORIZATION authid FOR PROGRAM name VERSION ver
```

Operand Definitions

Operand	Description
<i>authid</i>	One- to eight-character authorization ID that is prefixed to the program name to identify the new, alternate plan.
<i>name</i>	Name of the program for which the alternate plan is defined.
<i>ver</i>	Version number of the program for which the alternate plan is defined. For more information about valid versions to specify, see Using Version Clauses in CA Ideal Commands in the chapter "Preliminary Concepts."

DEFINE OUTPUT DESTINATION NETWORK Command

The DEFINE OUTPUT DESTINATION NETWORK command defines default settings for the specified network printer. The settings include:

- Printing a header and trailer page
 - Width of the print line
 - Initial and final formfeed control
 - Maximum number of prints per transaction
- These values override the actual printer settings.

CA Ideal supports network printers that are 328x compatible and that recognize the following printer control orders:

- **Forms Feed (FF)** : X'0C'
- **Carriage Return (CR)** : X'0D'
- **New Line (NL)** : X'15'
- **End of Message (EM)** : X'19'

For more information on FF opt and PERTASK *num* options, see the Network Printer Definition Considerations.

This command has the following format:

```

                                [WIDTH width ]
                                [HEADER opt  ]
DEFINE OUTPUT DESTINATION NETWORK dest [TRAILER opt ] ...
                                [FF opt     ]
                                [PERTASK num ]
    
```

Note: The syntax is positional. This allows you to specify multiple options, but you must specify them in the same order as shown in the previous display. For example, DEF OUT DEST NET xxxx HDR NO TRL NO is correct; but DEF OUT DEST NET xxxx TRL NO HDR NO results in a syntax error.

Operand Definitions

Operand	Description
<i>dest</i>	Printer destination name.
WIDTH <i>width</i>	Width of the print line. For network printers the value can be 80-250. Specify 0 to reset the width to the default setting.
HEADER <i>opt</i>	Defines whether a header page prints. Specify YES to print a header page and NO to not print a header page. (Default is YES).
TRAILER <i>opt</i>	Defines whether a trailer page prints. Specify YES to print a trailer page and NO to not print a trailer page. (Default is YES).
FF <i>opt</i>	(Network printer only) Controls the PSS-generated formfeeds at the beginning and end of an output. YES Issues formfeed at the start and end of print. NO Suppresses formfeeds. HEADER Issues formfeed at the start of print. TRAILER Issues formfeed at the end of print.

Operand	Description
PERTASK <i>num</i>	(Network printer only) Specifies the maximum number of outputs that can process during one print transaction for a specified printer.

DEFINE OUTPUT DESTINATION SYSTEM Command

The DEFINE OUTPUT DESTINATION SYSTEM command establishes the output destination of a system printer (your local printer). You can define a system type output destination to CA Ideal that is actually a data set. You can direct CA Ideal reports to that data set through the ASSIGN REPORT command in a batch run. For example:

```
...
//arpt DD DSN=site.dataset.b1330,DISP=OLD
Syntax//SYSIN *
ASSIGN REPORT arpt TO DEST SYS b1330
RUN apgm
...
```

B1330 is defined to CA Ideal as a SYStem OUTput DESTination with a blocksize specified.

This command has the following format:

```
DEFINE OUTPUT DESTINATION SYSTEM name [BLKSIZE nnnnn]
```

Operand Definitions

Operand	Description
SYSTEM <i>name</i>	Specifies the one- to eight-character name of a system printer.
BLKSIZE <i>nnnnn</i>	(z/OS batch only) Specifies the maximum blocksize for output assigned to a data set when executing in batch. The default blocksize is 32 times the record length. Valid values are 1-32767 bytes. The actual blocksize is calculated based on the record length and equals the number of records that fits in the specified maximum blocksize.

DELETE Command

The DELETE command or equivalent DELETE prompter deletes the definition of an entity-occurrence in history or test status.

You can also delete a dataview definition for SQL, VSAM, or unmodeled sequential file dataviews. You cannot delete dataviews modeled in the dictionary in CA Ideal. They are maintained and deleted using CA Datacom/DB facilities. For more information on specifying versions, see Using Version Clauses in CA Ideal Commands in the “Preliminary Concepts” chapter.

This command has the following format:

```
      {DATAVIEW}  
      {REPORT  }  
DELETE {PANEL  } name VERSION nnn  
      {PROGRAM }  
      {SYSTEM  }  
      {USER    }
```

Operand Definitions

Operand	Description
<i>name</i>	Definition name: SQL dataview: One- to eight-character authorization ID followed by the 1- to 18-character dataview name, separated by a period in the form <i>authid.dvw-name</i> Unmodeled dataview: 1- to 18-character name that identifies the dataview REPORT, PANEL, or PROGRAM 1 to 8 characters SYSTEM 1- to 15-character system name or one- to three-character short ID USER 1- to 15-character user name or one- to three-character short ID
<i>nnn</i>	Number CA Ideal assigned when this version of the definition was created. To delete the production version of a definition, first mark it to history.

DELETE AUTHORIZATION Command (CA Datacom SQL Access)

The DELETE AUTHORIZATION command deletes a previously defined plan for a program that uses CA Datacom SQL access. The plan is identified by a name created from the authorization ID, program name, and version number specified in the DEFINE AUTHORIZATION command. You can enter this command as DELETE AUTHORIZATION, DELETE ATZ, DELETE AUT, or as DELETE AUTH. You must enter the command from the system that contains the program object.

The statement-IDs for each statement must match in every plan associated with the program or a runtime error occurs. This type of error can happen if the CA Datacom utility DDTRSLM exports a plan without its associated program or if the CA Ideal Object Transport Utility transports a program object without its associated plan.

This command has the following format:

```
DELETE AUTHORIZATION authid FOR PROGRAM name VERSION ver
```

Operand Definitions

Operand	Description
<i>authid</i>	One- to eight-character authorization ID.
<i>name</i>	Name of the program for which the alternate plan is deleted.
<i>ver</i>	Version number of the program for which the alternate plan is deleted. For more information about valid versions to specify, see Using Version Clauses in CA Ideal Commands in the chapter "Preliminary Concepts."

DELETE MEMBER Command

The DELETE MEMBER command or the equivalent DELETE prompter deletes a member.

This command has the following format:

```
DELETE [MEMBER mem-name [USER user-name]]
```

Operand Definitions

Operand	Description
<i>mem-name</i>	One- to eight-character member name.

<i>user-name</i>	One- to 15-character user name or the one- to three-character <i>user-id</i> . This clause is required only if the specified member belongs to another user.
------------------	--

Example

```
DELETE MEMBER SIGNON
```

DELETE MODULE Command

The DELETE MODULE command deletes the MODULE entity occurrence that corresponds to the specified program or panel from Datadictionary. This stops the RUN processor from using the module form of the specified program or panel. The next time the online environment is started or a batch job is run, the VLS form of the program or panels (if any) is used. However, if an online environment is active when the DELETE MODULE command is executed, the module form of the program continues to be used until a REFRESH command is executed. The DELETE MODULE command does not delete the generated modules from any system module libraries. That is the user's responsibility. Be sure to delete all three generated parts of a CA Ideal program module.

Execute a SELECT SYSTEM command before the DELETE MODULE command to identify the CA Ideal system that contains the program.

For more information on Delete module format, see the *Administration Guide*

This command has the following format:

```
                {PROGRAM pgm-name}  
DELETE MODULE [FOR] {PANEL pnl-name }
```

Operand Definitions

Operand	Description
<i>pgm-name</i>	Name of the CA Ideal program to delete.
<i>pnl-name</i>	Name of the CA Ideal panel to delete.

DELETE OUTPUT Command

The DELETE OUTPUT command removes output from the output library. Issuing the DELETE command without any operands displays the DELETE prompt.

This command has the following format:

```
DELETE OUTPUT {name }
              {number}
```

Operand Definitions

Operand	Description
<i>name</i>	One- to eight-character name of the output. The name must be unique; otherwise a message and a list of all outputs with that name appear.
<i>number</i>	One- to four-digit output number.

Example

```
DELETE OUT RPT23
DEL OUTPUT COMPLIST
DEL OUT 1234
```

DELETE OUTPUT DESTINATION Command

The DELETE OUTPUT DESTINATION command removes the output destination of either a system (SYSTEM) or network (NETWORK) printer.

This command has the following format:

```
DELETE OUTPUT DESTINATION {SYSTEM name }
                          {NETWORK name}
```

Operand Definitions

Operand	Description
SYSTEM <i>name</i>	Specifies the one- to eight-character name of the system printer.
NETWORK <i>name</i>	Specifies the one- to eight-character name of the network printer.

DELETE PACKAGE Command (DB2 Only)

The DELETE PACKAGE command or equivalent prompter deletes an existing package definition. This command deletes only the CA Ideal package definition. It has no effect on the CA Ideal application itself, on the SQL module created as part of the GENERATE PACKAGE process, or on the actual DB2 package if one was generated for this package definition.

To display the DELETE prompter, select Option 4 on the Plan Menu or type DELETE or DELETE PACKAGE.

This command has the following format:

```
DELETE PACKAGE [name]
```

Operand Definitions

Operand	Description
<i>name</i>	One- to seven-character name of a package definition.

DELETE PLAN Command (DB2 Only)

The DELETE PLAN command or equivalent prompter deletes an existing plan definition. Use to perform this action. This command deletes only the CA Ideal plan definition. It has no effect on the CA Ideal application itself, on the SQL modules created as part of the GENERATE PLAN process, or on the DB2 application plan if one was created for this plan definition.

To display the DELETE prompter, select Option 4 on the Plan Menu or type DELETE or DELETE PLAN.

This command has the following format:

```
DELETE PLAN [name]
```

Operand Definitions

Operand	Description
<i>name</i>	One- to seven-character name of a plan definition.

DEQUEUE Command

The DEQUEUE command releases an entity from shared or exclusive use. This command is useful after certain kinds of system failures when CA Ideal maintains an entity's status as currently being edited and, therefore, unavailable for other editing. Dequeuing might also be necessary when a system failure occurs while displaying an index, output, or jobcard.

Use this command with care and only when the administrator is certain that there is an erroneous enqueue. Do not use DEQUEUE to dequeue a resource enqueued from a different address space (z/OS) or partition (VSE).

For more information about enqueues, see "Error recovery tools" in the *Problem Determination Guide* and "Application Migration Considerations" chapter in the *Administration Guide*.

Important! If an entity is being edited when it is dequeued, library integrity can be lost.

This command has the following format:

```

[ {SYSTEM } ]
[ {USER } ]
[ {PROGRAM} name VERSION {PROD} [SYSTEM sys] ]
[ {PANEL } {nnn } ]
[ {REPORT } ]

[ LIBRARY libname ]
[ MEMBER name [USER id] ]

[ {Tnnn} ]
[ DATAVIEW name [VERSION {nnn } ] ]
DEQUEUE
[ OUTPUT {name } ]
[ {number} ]

[ PACKAGE pkg-name ]
[ PLAN plan-name ]

[ JOBCARD {user-id} ]
[ {MASTER } ]

```

User Considerations

Libname is the name of the library specified in an error message.

PRODUCTION is only valid in the VERSION clause for programs that were loaded at a target site with the CA Ideal Object Transport Utility. For more information, see the *Working in the Environment Guide*.

A VERSION clause is required for dataviews for CA Datacom CBS dataviews, modeled sequential file dataviews, unmodeled sequential file dataviews, and VSAM dataviews. It is not used for SQL dataviews. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the "Preliminary Concepts" chapter.

Operand Definitions

Operand	Description
<i>name</i>	One- to seven-character name of a plan definition.

DETAIL Command

The DETAIL command or equivalent PF key displays the detail definition component of the current report definition. The report detail definition fill-in, shown in the *Generating Reports Guide*, specifies the fields to appear in each detail line of the body of the report and specifies sorting, control breaks, summary functions, and so on, for these fields.

This command has the following format:

DETAIL

DISABLE Command

The DISABLE command forces existing users to exit from an active application to import a new version of that application by using the Object Transport utility or by creating a new version of a load module. You can use the DISABLE command against applications that are in either VLS object format or load module format. The DISABLE command must specify the highest level of an application (that program executed by a RUN command). It has no effect if issued for a subprogram. Precede the DISABLE command with the appropriate SELECT SYSTEM command. A prompter panel is provided for the DISABLE command. The prompter is accessed either from a menu selection or by entering the DISABLE command.

This command has the following format:

```
DISABLE RUN pgm-name [AT] hh:mm [TOMORROW]
      [[TERMINATION] PANEL pnl-name VERSION nnn SYSTEM sys-id]
      [[WARNING] MESSAGE 'msg-text' ]
```

Operand Definitions

Operand	Description
<i>pgm-name</i>	Identifies the highest-level program in the application to disable. Any user executing the commands SELECT SYSTEM <i>sys-ID</i> ; RUN <i>pgm-name</i> ... is affected by the DISABLE command. Only the PROD version of that application is disabled. For users currently running an application that was disabled, the user's run is aborted the first time he presses the Enter key or a PF key or the application transmits a panel after the time specified in the DISABLE command. When a run is disabled, the TERMINATION PANEL displays (either the user-specified panel or the default panel).
<i>hh:mm</i>	Specifies when the disable takes effect. The hh:mm parameter is specified in 24-hour clock format, for example, 1 PM is 13:00. If the TOMORROW option is not specified, the time must be greater than the current time. Note: You can specify a semicolon (;), comma (,) period (.) or HHMM instead of the colon (:).

Operand	Description
<i>TERMINATION clause</i>	<p>Specifies the panel sent to the user upon termination of a run due to a DISABLE command. After sending the panel, the next time that user presses Enter or a PF key, the user is signed off of CA Ideal or returned to the CA Ideal main Menu, depending on the user's current setting of the SET RUN QUITIDEAL option. If no TERMINATION PANEL is specified, a default panel is sent that says: "The current application has been disabled by site management. Please press ENTER to continue."</p> <p>If specified, the TERMINATION clause must specify the panel name, the CA Ideal system it is defined in, and its numeric version number. The specified panel can have help, prefix, and suffix panels defined to it. The advantage of a user specified panel is that instructions can display telling the user when to sign back on.</p>
<i>'msg-text'</i>	<p>A message that displays in the message line for any user running the application. The maximum length of the message is as many characters as fit on the command line, on the line of a member being executed, or in the prompting field for the command.</p> <p>If a user starts running the program before the "DISABLE time", this message is sent at each transaction boundary (when he presses a PF key or the Enter key or when a panel is transmitted) to notify the user to sign off the application. If no WARNING MESSAGE is specified, the message sent is:</p> <p>This application will terminate at mm:dd hh:ss. Current time is hh:ss.</p> <p>Users executing a RUN command after the DISABLE command is executed, but before the "DISABLE time" arrives, are allowed to start running, but get this warning message. The warning message can be overwritten by system messages or messages sent through the NOTIFY statement.</p>

Important User Considerations

If the application to disable is running in multiple CICS regions or partitions, you must execute the DISABLE command from a session in each CICS. This is true for each “stand-alone” CICS region and each MRO region.

When the “DISABLE time” arrives, the users running the specified application are not immediately purged. Their runs are aborted at the next transaction boundary (when they press a PF key or the Enter key or when a panel is transmitted) after the disable time arrives. With programs and panels in VLS format, the enqueues on those programs and panels are not released until the run is actually aborted. If a user left the terminal before the disable time arrives, those enqueues are not released until the user returns and presses the Enter key or a PF key.

These enqueues prevent the Object Transport utility from running or a new version of a program from being marked to PROD status, because they attempt an exclusive enqueue on those programs and panels. In this case, eliminating these enqueues requires you issue a DEQUEUE command for each program and panel enqueued before running the Object Transport utility or issuing the MARK STATUS command. This is not a problem for application programs and panels in load module format since the run-time executor does not use enqueues in this case.

The program name specified in the DISABLE command must be the same program name specified in the RUN command. If the user specifies a subprogram name instead of the RUN or main program name, no error message is produced, but the command is ignored.

If a new version of a program is replaced and it is used by multiple applications, each of those applications must be disabled. If a new version of a program is replaced without issuing a DISABLE command or in some way making sure that all users are off the application, different kinds of internal errors can occur. The most common error indicates that the updateable and reentrant portion of a program are not synchronized. If the updateable portion of a program is in CICS temporary storage while the reentrant portion of the program is replaced, the next time that part of a program is loaded into memory this synchronization error is detected.

The DISABLE command stays in effect until it is explicitly released with the ENABLE command or until CICS is shut down and restarted.

DISPLAY AUTHORIZATION Command

The DISPLAY AUTHORIZATION command shows the current CA Ideal authorizations for the site.

This command has the following format:

```
DISPLAY AUTHORIZATION OPTIONS
```

DISPLAY DATAVIEW Command

The DISPLAY command or equivalent DISPLAY prompter displays an existing dataview definition using

For CA Datacom CBS dataviews and modeled sequential file dataviews: You can produce a formatted display of a dataview definition, including field descriptions, if the dataview was defined to the Datadictionary and cataloged in CA Ideal.

For SQL, VSAM, and unmodeled sequential file dataviews: You can produce a formatted display of a dataview definition that was defined to the dictionary facility and cataloged in CA Ideal, or, before the dataview was cataloged, you can produce a listing as entered. For VSAM and unmodeled sequential file dataviews, entering the DISPLAY DATAVIEW command also makes the specified dataview the current CA Ideal entity.

You access the DISPLAY DATAVIEW prompted by selecting Option 1 on the Dataview Maintenance Menu or by issuing the DISPLAY DATAVIEW command.

This command has the following format:

```

DISPLAY {DATAVIEW {auth-id.dvw-name      }}
        {          {dvw-name VERSION version}}

        [IDENTIFICATION]
        [FIELDS          ]
        [PARAMETERS     ]
        [KEYS            ]
    
```

Operand Definitions

Operand	Description
*	Represents the current dataview. Can be substituted for the dataview identification phrase.
<i>auth-id</i>	One- to eight-character authorization ID required for SQL dataviews only.
<i>dvw-name</i>	One- to 18-character name of the dataview.

Operand	Description
<i>version</i>	Version of the dataview. For modeled dataviews (CA Datacom native access or sequential), the specified version must be in test or production status. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the "Preliminary Concepts" chapter. If the version is not specified, the default version set in the SET VERSION command is used for both modeled and unmodeled dataviews.
IDENTIFICATION FIELDS PARAMETERS KEYS	Specify one of these keywords to display the indicated fill-in panel of a VSAM or unmodeled sequential file dataview definition. If the dataview was cataloged, the catalog listing displays by default. If the dataview was not cataloged, the Field Definition displays by default. Note: KEYS is only valid for VSAM KSDS data sets

DISPLAY INDEX Command

The DISPLAY INDEX command lists the name and status of each occurrence of the specified entity type. Optionally, this list or index can include occurrences of entity types that are related to a given entity type or occurrence (for example, each panel related to a given program). DISPLAY INDEX without an entity type calls up a prompter. In batch, CA Ideal processes DISPLAY commands as prints. This display is based on dictionary facility relationships.

In relating dataviews to systems or relating programs to systems, authorization is required for the specified system.

See also the DISPLAY INDEX OUTPUT command.

This command has the following format:

Display Index for All PROGRAM Parameter

```
DISPLAY INDEX ALL [PROGRAM] [ent-name] [VERSION version]
```

```
  {RELATED [TO] {PANEL } [ent-name] VERSION version}
    {PROGRAM}
    {REPORT }
```

Display Index for MEMBER Entity Type

```

DISPLAY INDEX MEMBER [ent-name] [VERSION version] [USER user-id]

    [RELATED [TO] {DATAVIEW} [ent-name] [VERSION version]]
        {PANEL }
        {PROGRAM }
        {REPORT }
        {SYSTEM }
        {USER }
    
```

Display Index for PACKAGE and PLAN Entity Types

```

DISPLAY INDEX PACKAGE [ent-name] [VERSION version]

DISPLAY INDEX PLAN [ent-name] [VERSION version]
    
```

Display Index for DATAVIEW, PANEL, PROGRAM, REPORT, SYSTEM, and USER Entity Types

```

DISPLAY INDEX ent-type [ent-name] [VERSION version]

    [RELATED [TO] {DATAVIEW} [ent-name] [VERSION version]]
        {PANEL }
        {PROGRAM }
        {REPORT }
        {SYSTEM }
        {USER }
    
```

Operand Definitions

Operand	Description
ALL [PROGRAM]	<p>You can enter ALL only when <i>ent-type</i> is PROGRAM to display an index of all programs, regardless of system, related to a specified program.</p> <p>When you specify ALL PROGRAM, you must specify the RELATED clause with a unique <i>ent-name</i> of a panel, report, or program that exists in the current system.</p> <p>Because the ALL option displays a cross system program index, you must have an authorization of ADMIN to use this option.</p> <p>Note: The ALL operand requires that a version, where applicable, uniquely qualify the RELATED TO entity. It does not take a default.</p>

Operand	Description
<i>ent-type</i>	<p>The class of entities you want to index. Entity types are SYSTEM, USER, DATAVIEW, PROGRAM, PANEL, REPORT, MEMBER, PACKAGE and PLAN.</p> <p>Entity types PLAN and PACKAGE cannot be related to another entity type. In the RELATED TO clause, you cannot specify MEMBER, PLAN, and PACKAGE as an entity type.</p> <p>If <i>ent-type</i> in the RELATED clause is PROGRAM, the specified resources of each program display across all systems. For example, the command DISPLAY INDEX PNL RELATED TO PGM displays an index of all panels each program uses, regardless of the system that contains the panels.</p> <p>However, if the <i>ent-type</i> is PROGRAM for both clauses and ALL is not specified, such as DISPLAY INDEX PGM RELATED TO PGM, subprograms in other systems are listed for each program, but calling programs in other systems are not included.</p>
<i>ent-name</i>	<p>(Optional) The specific name of an entity occurrence to index. SQL dataview names must include the authorization ID (for example, AUTHID1.PAY). You can use the following characters in an entity name to mask characters in the name during the index search.</p> <p>Important! Mask characters do not work with members and plans, because members and plans are not Datadictionary controlled entities.</p> <p>An asterisk (*) marks the position in the entity name of a single character to mask. For example, PAY*1 matches PAY01, PAY21, and so forth.</p> <p>Note: When you use an asterisk in a string, you must know the length of the string to match. If the name being tested has more characters than the masked value, the name does not match. For example, neither PAY*1 nor PAY* match PAY011.</p> <p>A plus sign (+) marks the position in the entity name beyond which all characters are masked.</p> <p>For example, DISPLAY INDEX PROGRAM IDE+ matches all names starting with IDE of any length. This command locates both PROGRAM IDE1 and IDE11. However, if the asterisk mask character were substituted for the plus sign mask character, the program IDE1 is located, but not the program IDE11.</p> <p>You can use the mask characters in any position except the first position. The plus sign mask character must be in the last position specified. When you use mask characters to search for systems or users, use the full entity name, not the short ID.</p>

Operand	Description
VERSION <i>version</i>	Version of the entity occurrence. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the “Preliminary Concepts” chapter.
USER <i>user-id</i>	The 1- to 15-character user name or the one- to three-character user ID. Note: The USER operand is used only when MEMBER is specified as the first clause <i>ent-type</i> . A user ID is required for an index of members belonging to another user.
RELATED TO	When used with DISPLAY INDEX ALL PROGRAM the RELATED TO entity types are limited to PROGRAM, PANEL and REPORT. The ALL operand requires that a version, where applicable, uniquely qualify the RELATED TO entity. It does not take a default. The RELATED TO option is used to find the RESOURCEs (except dataviews) of programs. This information, as well as the dataview relationship, is also available from Datadictionary. DDUTILTY can be used to produce RELATIONSHIP reports in batch.

Displaying Related Programs

If you relate program entity type to program entity type, for example, DISPLAY INDEX PROGRAM MY-PROG RELATED TO PROGRAM, the index includes all subprograms called by the specified program (MY-PROG) and all programs that call the specified program. This applies to each program if more than one is indexed.

The following report illustrates the index of all programs in System DOC, produced by the command: DISPLAY INDEX PGM

IDEAL: DISPLAY INDEX		PGM		SYS: DOC		DISPLAY		
Command Name	Ver	S	R-S	Sys	Lang	Description	Created	Updated
===== T O P =====								
000001	DEM01	001	T	PRV	DOC	IDEAL produce demrpt1	11/04/04	11/04/04
000002	DEM02	001	T	PRV	DOC	IDEAL produce mult dw details	11/04/04	11/04/04
000003	DEM03	001	T	PRV	DOC	IDEAL produce demrpt3-xsys res	11/04/04	11/04/04
000004	DEM04	001	T	PRV	DOC	IDEAL produce demrpt4	11/04/04	11/04/04
000005	DEM05	001	T	PRV	DOC	IDEAL list for reporting	11/04/04	11/04/04
000006	DEM06	001	T	PRV	DOC	IDEAL produce dempgm6 fm dw	11/04/04	11/04/04
000007	DEM06	002	T	PRV	DOC	IDEAL produce dempgm6 fm dw	11/04/04	11/04/04
000008	TEST	001	T	PRV	DOC	IDEAL test program for doc	10/05/04	10/05/04
===== B O T T O M =====								

Field Definitions

Field	Description
Name	One- to eight-character entity name..
Ver	One- to three-digit number, QSAM or SAM identifier, or SQL label (DBSQL or DB2) CA Ideal assigned when this version of the entity was created.
S	Status version of the entity: T (test-status version) P (production-status) H (history-status version)
R-S	Run-status of the entity: PRV (Private) SHR (Shared) RES (Resident)
Sys	System where the entity belongs.
Lang	Language that creates this entity. IDEAL is the default. The values ASM, COBOL, or PLI identify Assembler, COBOL, and PL/I respectively.
Description	One- to 32-character description of the entity.
Created	Date when the entity was created.
Updated	Date when the entity was last accessed in edit mode.

Displaying Related Dataviews

The following illustrates the index of dataviews related to all programs in system DOC, produced by the command: DISPLAY INDEX DATAVIEW RELATED TO PROGRAM

```

.....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: DISPLAY INDEX          DW          SYS: DOC  DISPLAY

Command Name          Ver S U Type Description          Created Updated
=====
000001
000002 =====> Related to PGM DEMO1      Version 001 Status TEST
000003 CUSTOMERS      001 P N      ORDER ENTRY DATAVIEW      01/30/02 10/05/04
000004
000005 =====> Related to PGM DEMO2      Version 001 Status TEST
000006 CUSTOMERS      001 P N      ORDER ENTRY DATAVIEW      01/30/02 10/05/04
000007 ORDERS          001 P Y      ORDER ENTRY DATAVIEW      01/30/02 09/02/04
000008
000009 =====> Related to PGM DEMO3      Version 001 Status TEST
000010 CUSTOMERS      001 P N      ORDER ENTRY DATAVIEW      01/30/02 10/05/04
000011
000012 =====> Related to PGM DEMO4      Version 001 Status TEST
000013 CUSTOMERS      001 P N      ORDER ENTRY DATAVIEW      01/30/02 10/05/04
000014
000015 =====> Related to PGM DEMO5      Version 001 Status TEST
    
```

Field Definitions

Field	Description
Name	One- to eight-character entity name.
Ver	One- to three-digit number. CA Ideal assigned when this version of the entity was created.
S	Status of the entity: T (Test) P (Production) H (History)
U	Update the dataview records: Y (Yes) N (No)
Description	The 1- to 32-character description of the entity.
Created	Date when the entity was created.
Updated	Date when the entity was last accessed in edit mode.

DISPLAY INDEX Line Commands

The DISPLAY INDEX Line commands let you enter a number of line commands directly in the margin of the index display.

You can use the asterisk (*) line command with any entity type to scroll that line to the top of the display. (For more information, see the, "Editing Commands," chapter.) You can use other line commands with programs, panels, reports, systems, members, and dataviews, as appropriate. They perform the same functions available through primary commands, except as noted below. Due to the six-character width of the display margin area, you can only enter some commands in abbreviated form.

The following shows a sample index of programs with line commands to delete three programs.

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....
IDEAL: DISPLAY INDEX          PGM          SYS: DOC  DISPLAY
Command Name  Ver S R-S Sys Lang Description          Created Updated
=====
del001 DEMO#1  001 T PRV DOC IDEAL produce demrpt1      11/04/94 11/04/94
000002 DEMO1   001 T PRV DOC IDEAL produce demrpt1      11/04/94 11/04/94
000003 DEMO2   001 T PRV DOC IDEAL produce mult dw details 11/04/94 11/04/94
000004 DEMO3   001 T PRV DOC IDEAL produce demrpt3-xsys res 11/04/94 11/04/94
000005 DEMO4   001 T PRV DOC IDEAL produce demrpt4       11/04/94 11/04/94
000006 DEMO5   001 T PRV DOC IDEAL list for reporting     11/04/94 11/04/94
del007 DEMO6   001 T PRV DOC IDEAL produce dempgm6 fm dw  11/04/94 11/04/94
000008 DEMO6   002 T PRV DOC IDEAL produce dempgm6 fm dw  11/04/94 11/04/94
del009 TEST    001 T PRV DOC IDEAL test program for doc   10/05/94 10/05/94
=====
                                B O T T O M=====
    
```

The following table shows the commands that you can enter in the index display and the types of entities supported for each command.

Command	Index Type								
	PGM	PNL	RPT	SYS	USR	MEM	DVW	PLAN	MOD
CATalog							X		
COM, COMPILe	X								
DEB, DEBUG	X								
DEL, DELETE (1)	X	X	X	X	X	X	X	X	
DIS, DISPLAY	X	X	X	X	X	X	X	X	
DUP (2)	X	X	X	X	X		X		
EDI, EDIT	X	X	X	X	X	X	X	X	

PROD (3)	X	X	X	X	X				
HIST (4)	X	X	X	X	X				
PRI, PRINT	X	X	X	X	X	X	X	X	X
RUN	X								
SUB, SUBMIT							X		
*	X	X	X	X	X	X	X	X	X

- (1) You cannot delete dataviews that are modeled in the dictionary in CA Ideal.
- (2) DUP means duplication to NEXT VERsion only, since DUP...NEWNAME requires information (the new name) which you cannot enter on the display panel. You cannot duplicate modeled and SQL dataviews to the next version.
- (3) PROD is short for MARK STATUS...TO PROD. You cannot mark modeled and SQL dataviews to production status in CA Ideal.
- (4) HIST is short for MARK STATUS...TO HIST. You cannot mark modeled and SQL dataviews to history status in CA Ideal.

Processing Hierarchy

You can enter any number of line commands in one display. Multiple commands are handled in three passes, as follows:

First Pass: Commands that do not need to interact with the user. This includes all commands displayed in the previous table except EDIT, DISPLAY, RUN, DEBUG, CATALOG, and DUPLICATE.

Successful execution of these commands clears the line command and the Updated column of the index display indicates which command was processed. This indicator remains in the data area until another command is issued against the same entity.

Second Pass: DISPLAY, EDIT, CATALOG, and DUPLICATE.

The DISPLAY and EDIT commands replace the index display with the entity DISPLAY or EDIT panel. You can scroll the DISPLAY or EDIT panels; the index and all pending commands are kept intact provided

- You do not change from one program component to another. However, you can change from one panel or report component to another without losing the index display.
- You do not change the display mode by entering the EDIT * command from a DISPLAY panel or the DISPLAY * command from an EDIT fill-in.
- You do not change entities when entering the NEXT entity command. For example, do not enter the NEXT RPT command from a program display.

The CATALOG process is followed by a DISPLAY of the dataview. The DUPLICATE process is followed by an EDIT of the new version. This means that DUPLICATE and CATALOG are both included in the second pass.

The processing of commands is interrupted after each EDIT or DISPLAY. You can either go on to the next pending display or edit or you can return to the index display:

- Use the NEXT entity command to change from one selected entity to the next. The word NEXT must be followed by the current entity type to distinguish it from the edit command NEXT. The common three-character abbreviations for entity types are allowed.

NEXT does not redisplay the index. Processing proceeds directly to the next selected entity display. If no more selected entities remain in the index, NEXT operates just like RETURN, showing the index display again.

- Issue the RETURN command to refresh the index display panel and reshow all pending commands.

Third Pass: RUN, DEBUG, or *

You can execute only one RUN, DEBUG, or asterisk (*) command. It is executed following all other pending commands. Following a RUN or DEBUG, the index display is lost when the command completes executing. If a RUN or DEBUG command is executed, the asterisk (*) command is not executed.

Example

In the first sample index display, DEL commands delete three programs.

```

.....+....1.....+....2.....+....3.....+....4.....+....5.....+....6.....+....7.....+....
IDEAL: DISPLAY INDEX          PGM          SYS: DOC  DISPLAY

Command Name  Ver S R-S Sys Lang  Description          Created  Updated
=====
del001 DEMO#1  001 T PRV DOC IDEAL produce demrpt1     11/04/04 11/04/04
000002 DEM01   001 T PRV DOC IDEAL produce demrpt1     11/04/04 11/04/04
000003 DEM02   001 T PRV DOC IDEAL produce mult dw details 11/04/04 11/04/04
000004 DEM03   001 T PRV DOC IDEAL produce demrpt3-xsys res 11/04/04 11/04/04
000005 DEM04   001 T PRV DOC IDEAL produce demrpt4      11/04/04 11/04/04
000006 DEM05   001 T PRV DOC IDEAL list for reporting    11/04/04 11/04/04
del007 DEM06   001 T PRV DOC IDEAL produce dempgm6 fm dw  11/04/04 11/04/04
000008 DEM06   002 T PRV DOC IDEAL produce dempgm6 fm dw  11/04/04 11/04/04
del009 TEST    001 T PRV DOC IDEAL test program for doc   10/05/04 10/05/04
=====
B O T T O M=====

```

In the resulting display, note the DELETED indicators in the Updated column. The user enters three commands: RUN, COM (Compile), and DIS (Display).

```

.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: DISPLAY INDEX          PGM          SYS: DOC  DISPLAY

Command Name  Ver S R-S Sys Lang  Description          Created Updated
=====
000001 DEMO#1  001 T PRV DOC IDEAL produce demrpt1     11/04/04 DELETED
000002 DEMO1   001 T PRV DOC IDEAL produce demrpt1     11/04/04 11/04/04
run003 DEMO2   001 T PRV DOC IDEAL produce mult dw details 11/04/04 11/04/04
com004 DEMO3   001 T PRV DOC IDEAL produce demrpt3-xsys res 11/04/04 11/04/04
000005 DEMO4   001 T PRV DOC IDEAL produce demrpt4       11/04/04 11/04/04
000006 DEMO5   001 T PRV DOC IDEAL list for reporting    11/04/04 11/04/04
000007 DEMO6   001 T PRV DOC IDEAL produce dempgm6 fm dw  11/04/04 DELETED
dis008 DEMO6   002 T PRV DOC IDEAL produce dempgm6 fm dw  11/04/04 11/04/04
000009 TEST    001 T PRV DOC IDEAL test for doc         11/04/04 DELETED
=====
B O T T O M=====
    
```

The compile is initiated and, in the next screen, the program DEMO6 displays.

```

.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION PGM DEMO6 (002) TEST          SYS: DOC  DISPLAY

Command.....1.....2.....3.....4.....5.....6.....7..
=====
000100 FOR FIRST 100 CUSTOMERS NO UPDATE
000200   PRODUCE DEMRPT6
000300 ENDFOR
=====
B O T T O M=====
    
```

Entering the RETURN command (PF2) returns the index display. The message line indicates that the compile completed and the Updated column for program DEMO3 shows COMPILE. The RUN is still pending until the user presses the Enter key.

```

IPCMA5YD10I- Compile for PGM DEMO3 successful; listing in OUTPUT 30
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: DISPLAY INDEX          PGM          SYS: DOC  DISPLAY

Command Name  Ver S R-S Sys Lang  Description          Created Updated
=====
000001 DEMO#1  001 T PRV DOC IDEAL produce demrpt1     11/04/04 DELETED
000002 DEMO1   001 T PRV DOC IDEAL produce demrpt1     11/04/04 11/04/04
RUN003 DEMO2   001 T PRV DOC IDEAL produce mult dw details 11/04/04 11/04/04
000004 DEMO3   001 T PRV DOC IDEAL produce demrpt3-xsys res 11/04/04 COMPILE
000005 DEMO4   001 T PRV DOC IDEAL produce demrpt4       11/04/04 11/04/04
000006 DEMO5   001 T PRV DOC IDEAL list for reporting    11/04/04 11/04/04
000007 DEMO6   001 T PRV DOC IDEAL produce dempgm6 fm dw  11/04/04 DELETED
000008 DEMO6   002 T PRV DOC IDEAL produce dempgm6 fm dw  11/04/04 11/04/04
000009 TEST    001 T PRV DOC IDEAL test for doc         11/04/04 DELETED
=====
B O T T O M=====
    
```

When you press the Enter key, program DEMO2 executes. The program index is no longer available. CA Ideal returns to the Main Menu, as shown below.

```

1-IDADRUNP04I - Run completed, RC = 0, OUTPUT = 356
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: MAIN MENU          PGM DEMO2 (001) TEST          SYS: DOC          MENU

Enter desired option number ==>          There are 11 options in this menu:

1. PROGRAM          Define and maintain programs
2. DATAVIEW        Display dataview definitions
3. PANEL            Panel Definition Facility
4. REPORT           Report Definition Facility
5. PLAN             Application Plan Maintenance
6. PROCESS          Compile, Run, Submit, Debug
7. DISPLAY          Display Entities
8. PRINT            Print Entities
9. ADMINISTRATION  Administration functions
10. HELP            Overview of HELP facilities
11. OFF             End IDEAL Session

```

Error Processing

If an error is encountered during command processing, the next command is suspended. The margin area is highlighted and the cursor is placed there. An error message remains at the top of the screen unless cleared. If the command prepared a display (such as CATALOG), that screen displays. To get back to the index display, issue the RETURN command or use the F2 key. Do not use the NEXT command as it only tries to re-execute the same command and encounter the same error again.

DISPLAY INDEX MODULE Command

The DISPLAY INDEX MODULE command determines which application programs and panels were converted into module format. You can also specify an optional WITH VERIFICATION clause to verify that the modules are really in the module library and represent the appropriate application program.

This command has the following format:

```

                                [ {VERIFY } ]
DISPLAY INDEX [ALL] MODULE [[WITH] {VERIFICATION}]

```

Operand Definitions

Operand	Description
DISPLAY INDEX	Displays or prints an index of all the programs and panels in the current system that were converted to module format, based on the MODULE entities in Datadictionary. The module name, the corresponding application program or panel identification, and the date/time of program compilation display. The entries are in alphabetical order by module name with all program modules first, followed by panel modules.
ALL	Gives an index listing of all programs and panels in all systems that were converted to module format. The order is alphabetical by module name. If you omit ALL, only modules for the current system display or print.
WITH VERIFICATION	Attempts to load each module or set of modules to determine whether the correct modules are available to the environment. The entity type, system ID, entity name, version number, and date/time stamp in the MODULE Datadictionary entity are compared with the information found in the loaded module. Any discrepancies are noted.

DISPLAY INDEX OUTPUT Command

The DISPLAY INDEX OUTPUT command lists the name and status of each output member for the current user. It is equivalent to DISPLAY OUTPUT STATUS.

This command has the following format:

```
DISPLAY INDEX OUTPUT
```

DISPLAY INDEX SESSION Command

The DISPLAY INDEX SESSION command lists the name and status of each current user. Information is provided to determine who is active in the Session Control Facility (SCF) environment. Computer Associates products that run in the SCF environment are CA Ideal and Datadictionary (DDOL). You can only issue this command under CICS in the VSE and z/OS environments.

The command returns a snapshot of the environment. The environment can change before the information is returned. The information that is returned is information retrieved from control blocks kept in CICS Auxiliary Temporary Storage records, which are updated at a transaction boundary. This implies that the information is not current if the user is an active CICS task.

If your terminal is split into multiple regions, a set of information is returned for each region. If you are running a program in the CA Ideal environment (which can only occur in region one), the main program, system, version, and the number of programs in the run unit are returned. (You can issue the RUN command only in region one.)

The IDs returned are the op-IDs defined for the user in the CICS signon table (SNT). If the signon table is not used, they are the IDs placed in the TCTTEOI field of the terminal control table (TCT). All user IDs are returned, regardless of the products the user is logged onto.

If you initiate an asynchronous task (a compile or network print) and log off before the completion message is sent, you still show up as logged on.

This command has the following format:

```
DISPLAY INDEX SESSION
```

The following example illustrates the index of users this command produces. Each column is described.

Command	User	OPID	Reg	User name	Term	Product	Sys	Main PGM	Ver	Sub
000001	RAY	RAY	1	RAY	Z000	IDEAL	WOR	WORUN	PRD	0017
000002	MDF	MDF	1	MDUFFY	Z002	IDEAL	WOR	WORUN	PRD	0018
000003	SFT	SFT	1	SEIFERT	Z003	IDEAL		**NONE**		0000
===== B O T T O M =====										

Field Definitions

Field	Description
User	One- to three-character CA Ideal user ID.
OPID	One- to three-character CICS op-ID for the user. This is the ID from the TCTTEOI (operator ID) field of the CICS TCTTE control block.
Reg	Terminal display region.
User name	The 1- to 15-character CA Ideal user name.
Term	Terminal ID.
Product	Name of the product used.
Sys	System where the main program resides, if a program is run.
Main PGM	Name of a program executed using a RUN command. If the first region is in CA Ideal but is not executing a program, **NONE** is returned.
Ver	Version of the main program, if a program is run.
Sub	Number of programs in the run unit for the Main Program. The total is all of the programs defined in the resource section of all the programs that were called. If a program is released, the storage is released but the program is still counted in the total.

DISPLAY JOBCARD Command

Each CA Ideal user is provided with a special fill-in called JOBCARD that they can edit to contain all JOB record information. Each site has a master JOBCARD fill-in. The JOBCARD member contains a maximum of five lines. CA Ideal uses the user's jobcard member automatically when a PRINT command is issued online with a destination of a system printer (resulting in a batch job submission).

This command has the following format:

```

[MASTER      ]
[             ]
DISPLAY JOBCARD [ {user-id } ]
[USER {user-name}]
    
```


Operand Definitions

Operand	Description
MASTER	Displays the master JOBCARD. If you omit this keyword, the user's JOBCARD displays.
<i>user-id</i> <i>user-name</i>	One- to three-character ID or the 1- to 15-character user name of the user who owns the JOBCARD. This clause is required if the JOBCARD does not belong to the current user.

DISPLAY LIBRARY STATUS Command

The DISPLAY LIBRARY STATUS command displays information about the VLS libraries in use. The source, object and panel information are for libraries related to the current CA Ideal SYSTEM.

Example

	Library Name	Blocks Allocated	Blocks In Use	Blocks Available	Per cent Full
System source library	ID\$IDSRC	8176	6527	1649	79
System object library	ID\$IDOBJ	5392	5204	188	96
System panel library	ID\$IDPNL	1048	217	831	20
Dataview library	IDDW	1840	679	1161	36
Member library	IDDAT	4186	1258	2928	30
IDEAL panel library	ADRPNL	1948	1762	186	90
IDEAL message library	ADRLIB	1396	556	840	39
IDEAL output library	ADROUT	1012	40	972	3

DISPLAY LMT Command

The DISPLAY LMT command displays the content of the Load Module Table, which may be a composite of entries loaded from Application Module Tables, and entries retrieved from the dictionary.

The entire table is displayed, regardless of the currently selected system. The entries are shown in the order they are found in the table, and are not sorted. If entries have been added by use of the REFRESH command, they will have been added at the end. See the FIND and INCLUDE commands in *Section 3* for help in locating individual entries.

This command has the following format:

DISPLAY LMT

Example

```
PGM $ID CICSWEB (PRD) IS MODULE CICSWEB
PGM CTH A8842296 (PRD) IS MODULE ADAM1
PGM CTH B8842296 (PRD) IS MODULE ADAM2
PGM CTH A7698144 (PRD) IS MODULE A769814
PGM CTH B7698144 (PRD) IS MODULE B769814
PGM CTH C1198608 (PRD) IS MODULE CM98608
PGM QAT CUST (PRD) IS MODULE CUST
PGM CTH C9370687 (PRD) IS MODULE C937068
PGM CHE DATE (PRD) IS MODULE D1
PGM SOL ID#2192 (PRD) IS MODULE ID#2192
PGM SOL ID#2773 (PRD) IS MODULE ID#2773
PGM SOL ID#2818 (PRD) IS MODULE ID#2818
PGM SOL ID#2854 (PRD) IS MODULE ID#2854
PGM SOL ID#2897 (PRD) IS MODULE ID#2897
PGM CTH C8668350 (PRD) IS MODULE LARRYH
PGM CHE MARKGEN (PRD) IS MODULE MARKGEN
PGM OBR PERSYS (PRD) IS MODULE OBPERSY
PGM JUL PERSYS (PRD) IS MODULE PERSYS
PGM CTH A8619139 (PRD) IS MODULE PGMA
PNL CTH P8842296 (002) IS MODULE ADAMZ
PNL CHE DATEPNL (001) IS MODULE D1
```

Note: The name after "IS MODULE" is the base seven-character name that was used on the CREATE MODULE command. The program entries in CICS will have an additional suffix character. It is possible for a program and a panel to have the same module name, as there is a prefix on the dictionary name, and different suffixes on the PPT entries.

DISPLAY MEMBER Command

The DISPLAY MEMBER command displays a member. DISPLAY MEMBER alone produces a prompter.

This command has the following format:

```
DISPLAY MEMBER [* ]
                [name [USER user-id] ]
```

Operand Definitions

Operand	Description
*	An asterisk represents the current member.
<i>name</i>	One- to eight-character member name.
<i>user-id</i>	The 1- to 15-character user name or the one- to three-character user-id required. This clause is required only if the member belongs to another user.

Example

```
DISPLAY MEMBER SIGNON
```

DISPLAY OUTPUT Command

The DISPLAY OUTPUT command displays an output that resides in the output library. Once you display an output with this command, you can scroll and view it using the various “browsing” commands described in Chapter 4. After the browsing activity is ended (by initiating another activity or entering the END command, the output is deleted (if the disposition is RELEASE). You can retain the output by entering the command KEEP OUTPUT before ending the browse.

This command has the following format:

```
          {name }
DISPLAY OUTPUT {number}
```

Operand Definitions

Operand	Description
<i>name</i>	Displays the output with the specified one- to eight-character name. The name must be unique; otherwise a message and a list of all outputs with that name appear.
<i>number</i>	Displays the output with the specified one- to four-digit number.

Example

```
DISPLAY OUTPUT COMPLIST
```

```
DISPLAY OUTPUT 1234
```


Field	Description
LINES	Network printer setting for the number of lines to print per page.
FF	Controls the PSS-generated form feeds at the beginning and end of an output. YES Issues form feed at the start and end of print. NO Suppresses form feed. HEADER Issues form feed at the start of print. TRAILER Issues form feed at the end of print.
PERTASK	Specifies the maximum number of outputs that can process during one print transaction for a specified printer.

DISPLAY OUTPUT STATUS Command

The DISPLAY OUTPUT STATUS command displays or edits the status of outputs in the output library.

This command has the following format:

```

                [OWN  ]
DISPLAY OUTPUT [ALL  ] STATUS
                [name ]
                [number]

```

Operand Definitions

Operand	Description
OWN	<i>(Optional)</i> Displays the status of outputs under the user's identification. DISPLAY OUTPUT STATUS with no operands obtains the same display.
ALL	Displays the status of all outputs.
<i>name</i>	Displays the status of the output with the specified one- to eight-character name. If the name is not unique, a warning message and a list of all outputs with that name appear.
<i>number</i>	Displays the status of the output with the specified one- to four-digit number.

On the output status display, you can position the cursor on the line showing an output and type a command to perform one of the functions shown in the following table.

Command	Meaning
Z	Delete (more than one allowed on the screen)
D or S	Display (only one allowed on the screen)
H	Alter disposition to Hold
K	Alter disposition to Keep
L	Alter disposition to Leave
R	Alter disposition to Ready
P	Reissue print for output already assigned to system or network print

Example

An example of the display that the DISPLAY OUTPUT ALL STATUS command produces follows:

Actions: D=display, Z=delete, P=print												
A	NUM	UID	NAME	CP	RT	DISP	DATE_CRE	TIME	DESCRIPTION	NRECS	DST	DST_NAME
	9	MMM	COMPLIST	01	02	READY	4/20/04	0933	TESTIT	56	LIB	
	11	KRI	COMPLIST	01	02	READY	2/03/04	1136	IQ936	98	LIB	
	20	CAC	IQ936	01	02	READY	12/12/04	1346	LIST STATEME	108	SYS	LPT1

Field Definitions

Field	Description
A	Column where you can place edit commands.
NUM	Number of the output.
UID (USER ID)	User identification.

Field	Description
NAME	<p>Name of the output. The name must be unique; otherwise a message and a list of all outputs with that name appear. The defaults are:</p> <p>COMPILE Name of the program when the destination is the output library or the default constant COMPLIST when the destination is a system or network printer..</p> <p>RUN The name depends upon the output generated; for example, if the program produces a report, the name is the name of the report.</p> <p>PRODUCE Name of the report.</p> <p>LIST Result of a LIST statement in a CA Ideal PDL procedure is a listing. The default output name is the program name when the destination is the output library or the default constant RUNLIST when the destination is a system or network printer..</p> <p>PRINT Name of the entity when the destination is the output library or the default constant PRTLST when the destination is a system or network printer..</p> <p>PRINT SCREEN PRSCREEN always displays in the NAME column.</p> <p>DEBUG The default output name for DEBUG LIST and ECHO output is DBUGLIST.</p>
CP (COPIES)	Number of copies to print on a system or network printer.
RT (RETENTION TIME)	Retention time in days before the output is eligible to be automatically deleted.
DISP (DISPOSITION)	<p>Disposition of the output.</p> <p>CRTIN Output in the output library is being created.</p> <p>PRINT Output in the output library is being printed.</p> <p>READY Output is available for print or display (see the <i>Working in the Environment Guide</i> for further information).</p> <p>HOLD Output is held (see the <i>Working in the Environment Guide</i> for more information).</p> <p>QHELD Output destination's queue is on hold.</p> <p>KEEP Output is ready to print with a copy retained in output library after printing (see the <i>Working in the Environment Guide</i> for more information).</p> <p>LEAVE Output was printed but was left in the output library at the request of the user using a KEEP disposition.</p> <p>PRINTD Output already printed but kept for further browsing and printing.</p> <p>DELET Output deleted.</p>

Field	Description
DATE_CRE	Date of the output request in a format specified at installation.
TIME (TIME REQUEST)	Time of the output request in 24-hour format.
DESCRIPTION	A 1- to 32-character description of the output. The following defaults apply if the user does not provide a description in the commands given following.

Command	Default description
COMPILE	COMPILE LISTING
LIST statement in PDL	LIST STATEMENT
PRINT	CA Ideal PRINT
PRINT SCREEN	Name of the screen being printed
PRODUCE	REPORT rptname
RUN	List or report, if one was generated as a result of the run, otherwise, no output is produced

The fields are defined as follows:

Field	Description
NRECS	Number of physical output lines in the report, excluding spacing and page skips.
DST (DESTINATION TYPE)	The possible destinations are: SYS System printer. LIB Output library. NET Network printer. MAIL [assign the value for emailp in your book] user.
DST_NAME (DESTINATION NAME)	Destination name.

DISPLAY PACKAGE Command (DB2 Only)

The DISPLAY PACKAGE command or equivalent prompter displays an existing package definition. To display the DISPLAY prompter, select Option 1 on the Plan Menu or type DISPLAY.

This command has the following format:

```

          [*          ] [IDE]
DISPLAY [PACKAGE [name]] [RES]
                               [PAR]

```

Operand Definitions

Operand	Description
*	Displays the current package.
name	One- to seven-character name of a package definition.
IDE	Displays the package identification fill-in.
RES	(Default) Displays the package resources fill-in.
PAR	Displays the package parameters fill-in.

DISPLAY PANEL Command

The DISPLAY command or equivalent prompter displays an existing panel definition using and makes it current.

Access the EDIT/DISPLAY prompter (as it applies to a panel definition) either by selecting Option 1 on the Panel Maintenance Menu or by issuing the DISPLAY Panel command.

You can display another component of the current panel definition by issuing the name of one of the panel definition components, NEXT or PREVIOUS, without DISPLAY.

This command has the following format:

```

          [          [          {nnn      }]]
[DISPLAY] [PANEL name [VERSION {PRODUCTION}]] [component]
          [          [          {LAST    }]]

```

Operand Definitions

Operand	Description
*	An asterisk represents the current panel.
<i>name</i>	One- to eight-character name of the panel definition. If omitted, the default is the current panel.
VERSION	Specific version of the panel. This entry is allowed (but not required) only if you specified a panel name. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the “Preliminary Concepts” chapter.

Note: If you omit both version and name, the default version of the current panel is used. If you specify name and omit version, the default version of the named panel is used.

Operand	Description
<i>component</i>	<p>One of the following:</p> <p>IDENTIFICATION Displays the panel identification definition fill-in.</p> <p>PARAMETER Displays the parameter definition fill-in.</p> <p>LAYOUT (Default) Displays the panel layout fill-in.</p> <p>SUMMARY Displays part of the panel layout in the top region of the screen (with each field numbered sequentially), and the field summary table in the bottom region of the screen..</p> <p>IRULES Displays the field input editing/validation rules table.</p> <p>ORULES Displays the field output editing rules table.</p> <p>FIELD id Displays the extended field definition of the specified field. The field can be identified by either name or number.</p> <p>FACSIMILE Displays the specified panel layout as it appears in its finished form.</p> <p>NEXT Displays the extended field definition for the field following the current field definition position.</p> <p>PREVIOUS Displays the extended field definition for the field that precedes the current field definition position.</p>

DISPLAY PCT Command

The DISPLAY PCT command displays information about PCT entries that reference SC00INIT, SC00DISP, SC00NATD, SC00SAST, and DFHWBA.

This command has the following format:

```
DISPLAY PCT
```

Example

An example of the display that the DISPLAY PCT command produces follows.

TRAN ID	INITIAL PROGRAM	TYPE OF ENTRY	TWA SIZE	PRD
scfd	SC00DISP	ACCOUNT-ID	64	
ASYN	SC00NATD	ASYN	64	WWW.ASYNC (0001)
CWBA	DFHWBA	WEB INTF.	0	Transaction not in SCWBTRAN
DDOL	SC00INIT		12	DDO
DDOX	SC00INIT		12	DDO
DEBB	SC00INIT		64	Transaction not in SC00TRAN
DEBG	DFHWBA	WEB INTF.	64	Transaction not in SCWBTRAN
DIAL	DFHWBA	WEB INTF.	64	Transaction not in SCWBTRAN
EMPL	SC00NATD	ASYN	64	Transaction not in SCASTRAN
HBRX	DFHWBA	WEB INTF.	0	Transaction not in SCWBTRAN
HBR1	SC00INIT	FINAL-ID	64	IDL \$ID.HBR1
HTTU	DFHWBA	WEB INTF.	0	Transaction not in SCWBTRAN
HWBA	DFHWBA	WEB INTF.	0	Transaction not in SCWBTRAN
IBIN	DFHWBA	WEB INTF.	64	WWW.WEBIMAGE(001)
IDBA	DFHWBA	WEB INTF.	64	WWW.BBSDEMOA(001)
IDBB	DFHWBA	WEB INTF.	64	WWW.BBSDEMOB(001)
IDBC	DFHWBA	WEB INTF.	64	WWW.BBSDEMOC(001)
IDB0	DFHWBA	WEB INTF.	64	WWW.BBSDEM00(001)
IDEA	SC00INIT		64	IDL
IDLX	SC00INIT	FINAL-ID	64	IDL
IPCV	SC00INIT		64	IPC
IPCX	SC00INIT		64	IPC
JULC	SC00NATD	ASYN	60	Transaction not in SCASTRAN
P08	SC00DISP	ACCOUNT-ID	64	
SAST	SC00SAST	COMP/PRINT	64	
SCFD	SC00DISP	ACCOUNT-ID	64	
S318	SC00NATD	ASYN	64	Transaction not in SCASTRAN

DISPLAY PCT operates by browsing the transaction entries and checking the program name returned. If it is one of SC00INIT, SC00DISP, SC00NATD, SC00SAST or DFHWBA, then the details are formatted for the display. The program name directly determines the TYPE OF ENTRY column, and the various types are discussed below:

ACCOUNT-ID

These are transactions that execute SC00DISP, and can be used as operands of the SET ENVIRONMENT ACCOUNT-ID command. The TWA size should be 64 bytes or more. No other details are shown.

FINAL-ID

These are transactions that execute SC00INIT, which starts an SCF session. They may execute CA Ideal, or CA Datacom applications, so there will be an entry in the PRD (product) column to indicate which is involved. This information is gathered from the entry in the SC00TRAN transaction table corresponding to the transaction id, and if none can be found, the message "Transaction not in SC00TRAN" will be displayed.

A CA Ideal application will also have a startup member, and the user and member names will be obtained from SC00TRAN and displayed. These transactions can be used as operands of the SET ENVIRONMENT FINAL-ID command, and should have a TWA size of 64 or more bytes. Transactions running other CA products may need only 12 bytes of TWA.

COMP/PRINT

There is usually only a single entry for transaction SAST, which executes SC00SAST, and is used in a CA Ideal development environment for running asynchronous compilations.

ASYNC

These are transactions executing SC00NATD. They are used to start CA Ideal applications that do not run at a terminal, such as those servicing a message queue. The transaction ID is looked up in the SCASTRAN table to find the system, name and version of the CA-Ideal program that will be run. If the entry is found those details are displayed. If not found, the message "Transaction not in SCASTRAN" is displayed. As with any other CA Ideal transaction, the TWA size should be 64 bytes or greater.

WEB INTF.

These transactions execute DFHWBA, the alias driver for the CICS Web Interface, and it may then call web applications in any language. You can expect many entries to show as "Transaction not in SCWBTRAN" which is the lookup table for web applications written in CA Ideal. In particular, the DEBG and DIAL transactions are special cases that do not need entries in the table, even though they do execute CA Ideal code.

Where an entry in SCWBTRAN does exist, the system, name and version of the program that will run is displayed. If the version shows as "PRD", then the current production version will always be executed. If a numeric version is shown, then that program must be in TEST status to execute.

DISPLAY PLAN Command (DB2 Only)

The DISPLAY PLAN command or equivalent prompter displays an existing plan definition using. To display the DISPLAY prompter, select Option 1 on the Plan Menu or type DISPLAY.

This command has the following format:

```

          [*          ] [IDE ]
DISPLAY [PLAN [name]] [RES ]
                               [DBRM]
                               [PARM]
                               [PKL ]

```

Operand Definitions

Operand	Description
*	Displays the current plan.
<i>name</i>	One- to seven-character name of a plan definition.
IDE	Displays the plan identification fill-in.
RES	(Default) Displays the plan resources fill-in.
DBRM	Displays the plan DBRM fill-in.
PARM	Displays the plan parameters fill-in.
PKL	Displays the plan package list fill-in.

DISPLAY PROGRAM Command

The DISPLAY PROGRAM command or equivalent prompter displays any component of an existing program definition and makes that definition the current definition. For non-ideal programs, you can only display or edit the identification and the parameter fill-ins.

You can specify the name of a component (or press the equivalent function key) to locate another component in the current program definition without specifying DISPLAY.

This command has the following format:

```

[ * ]
DISPLAY [PROGRAM name [VERSION version]]
[ENVIRONMENT ]
[IDENTIFICATION]
[RESOURCES ]
[PARAMETER ]
[WORK ]
[PROCEDURE ]
    
```

Operand Definitions

Operand	Description
DISPLAY	Calls up a prompter.
*	Displays the current plan.
<i>name</i>	One- to eight-character name of the program definition.
<i>version</i>	Specific version of the program. This entry is allowed (but not required) only if you specified a program name. If you omit both version and name, the default version of the current program is used. If you specify name and omit version, the default version of the named program is used (see the SET VERSION command for more information on setting the default version)
ENVIRONMENT	(CA Datacom SQL only) Displays the program environment definition fill-in. Program environment data is defined and maintained using the fill-in shown in <i>Creating Programs</i> . You can specify default values for the fill-in using the SET DBSQL command.
IDENTIFICATION	Displays the program identification definition fill-in.

Operand	Description
RESOURCES	Displays the program resources fill-in. The program resources fill-in specifies the resources that the application uses. These resources can include dataviews, panels, reports, and subprograms.
PARAMETER	Displays the parameter definition fill-in. Parameter data consists of the names and descriptions of data items to pass to this program from a calling program or to this program using a RUN command.
WORK	Displays the working data definition fill-in.
PROCEDURE	(Default) Displays the procedure definition fill-in. If you enter this command before the procedure is defined, a blank screen appears. If you enter this command after a procedure is defined, then as many lines of the procedure (from the top) as fit in the region display.

DISPLAY REPORT Command

The DISPLAY command or equivalent prompter displays an existing report definition.

You can specify the name of a component (or press the equivalent function key) to locate another component in the current report definition without specifying DISPLAY.

This command has the following format:

```
[ * ]
DISPLAY [REPORT name [VERSION version]]
      [IDENTIFICATION]
      [PARAMETER ]
      [HEADING ]
      [COLUMN ]
      [DETAIL ]
```

Operand Definitions

Operand	Description
*	Displays the current report.
<i>name</i>	One- to eight-character name of the specified report definition.

Operand	Description
<i>version</i>	Specific version of the program. This entry is allowed only if a report name was specified. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the "Preliminary Concepts" chapter.
IDENTIFICATION	Displays the report identification definition fill-in.
PARAMETERS	Displays the report parameters fill-in.
HEADING	Displays the report page heading definition fill-in.
DETAIL	(Default) Displays the report detail definition fill-in.
COLUMN	Displays the report column headings definition fill-in.

DISPLAY SESSION OPTIONS Command

The DISPLAY SESSION OPTIONS command displays the values of parameters set for the current session. You can display every option or a specified option. It includes those options set in the signon procedure, set during the current session, and the installation defaults.

For a complete list of the parameters that display for each option, see the *Working in the Environment Guide*. See this guide for the SET commands used to change parameters.

This command has the following format:

```
DISPLAY SESSION {OPTIONS }
                {option }
```

Operand Definitions

Operand	Description
OPTIONS	Display all options.
<i>option</i>	Specific session option to display. Options are:
\$RC	ENVIRONMENT
ASSIGN	OUTPUT
AUTHORIZATION	PANEL
CATALOG	PLAN
COMMAND	PROGRAM

Operand	Description
COMPILE DATA	REPORT
DATAVIEW	RUN
DBID	SCROLL
DBSQL	VERSION
EDIT	

DISPLAY SYSTEM Command

The DISPLAY command or equivalent prompter displays an existing system definition and makes it the current entity.

This command has the following format:

```

      [ *
DISPLAY [ {sname}
          [SYSTEM {sid } VERSION version ]

```

Operand Definitions

Operand	Description
*	You can substitute an asterisk for the entity type and entity name when the current entity is a system.
<i>sname</i>	The 1- to 15-character CA Ideal system name. (The system name is the name of the system definition.)
<i>sid</i>	Three-character system identifier of the system definition.
VERSION <i>version</i>	For more information on specifying valid versions, see Using Version Clauses in CA Ideal Commands in the chapter "Preliminary Concepts."

DISPLAY USER Command

The DISPLAY command or equivalent prompter displays an existing user definition using and makes it the current entity. This command has the following format:

```

      [*
DISPLAY [USER name [VERSION version]]

```

Operand Definitions

Operand	Description
*	You can substitute an asterisk for the entity type, entity name, and version for the current user.
<i>name</i>	The 1- to 15-character user name or the one- to three-character user ID of the user definition.
<i>version</i>	For more information about valid versions to specify, see Using Version Clauses in CA Ideal Commands in the chapter "Preliminary Concepts."

For more information about creating and maintaining user definitions, see the *Administration Guide*.

DUPLICATE Command

The DUPLICATE command or equivalent DUPLICATE prompter copies an existing definition to a new definition using.

After DUPLICATE performs successfully, the new version of the definition becomes the current definition and you see an identification fill-in. The content of the newly created version is identical to the previous version, only with a new name or version number. It is presented for modification. You can also modify this new definition at subsequent sessions, as long as it remains in test status.

This command has the following format:

```
[ * ]
[ { DATAVIEW} ]
[ { PANEL } ]
DUPLICATE [ { PROGRAM } name [VERSION version] ]
[ { REPORT } ]
[ { SYSTEM } ]
[ { USER } ]
[SYSTEM sid]
[NEWNAME newname]
[TO] [NEXT [VERSION]]
```

Operand Definitions

Operand	Description
*	You can substitute an asterisk for the entity type, name, and version for the current entity occurrence.
<i>name</i>	Definition to duplicate: <ul style="list-style-type: none"> ■ 1- to 18-character name of an unmodeled (sequential or VSAM file) dataview. ■ One- to eight-character program, report, or panel name ■ 1- to 15-character system name or user name ■ One- to three-character system ID or user ID
<i>version</i>	Version of the entity to duplicate. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the “Preliminary Concepts” chapter.
Operand	Description
SYSTEM <i>sid</i>	Three-character system short identifier for the program, panel, or report to duplicate. If you do not enter an ID, the current system is the default. Note: This option applies to PANEL, PROGRAM, and REPORT. You cannot specify the SYSTEM parameter with DUPLICATE USER, SYSTEM or DATAVIEW.
<i>newname</i>	Name the definition copies to. The keyword NEWNAME is required. Note: You cannot use NEWNAME with SYSTEM or USER. This is because you cannot duplicate a user or system to a new name because one of the attributes is a short-id that is also an alias for the entity. Duplicating results in a duplicate alias, which Datadictionary does not allow. You can duplicate these entities only to a new version.
NEXT	Copies the definition to the next version of that definition (default).

Example

Assume that a panel definition named UPDPNL, version 2, is copied to a new panel definition called UPTD. To do this, enter:

```
DUPLICATE PANEL UPDPNL VERSION 2 NEWNAME UPDT
```

The result is version 1 of a panel definition named UPDT in test status. To create the next version of an existing panel definition, enter:

```
DUPLICATE PANEL UPDPNL VERSION 2 NEXT VERSION
```

The result is version 3 of panel UPDPNL in test status in the current system. To create the next version of the current definition, you can shorten the command to DUP *. For example, if the current definition is version 2 of the panel UPDPNL, the results of the following two statements are the same.

```
DUPLICATE PANEL UPDPNL VERSION 2 NEXT VERSION
DUP *
```

DUPLICATE MEMBER Command

The DUPLICATE MEMBER command copies an existing member to a new member. If you enter the DUPLICATE command with no operands, a prompter displays that provides the complete syntax so you can complete the command.

After successful completion of the DUPLICATE command, the new member is now the current member and you see the EDIT/DISPLAY member fill-in. You can also modify the new member at subsequent sessions.

This command has the following format:

```

      { *           [ {username} ] }
DUPLICATE {MEMBER name [USER [user-id ] ] }
          [TO] NEWNAME newname [DESCRIPTION 'string']
```

Operand Definitions

Operand	Description
*	You can substitute an asterisk for the current member name.
<i>name</i>	One- to eight-character name of the member to duplicate.
<i>username</i>	The 1- to 15-character CA Ideal user name.

<i>user-id</i>	One- to three-character ID of the user who owns the member. This clause is required if the member does not belong to the current user.
<i>newname</i>	Name to give to the new member. This name must not be the name of an existing member belonging to the current or specified user.
DESCRIPTION ' <i>string</i> '	The 1- to 24-character descriptive text to store along with the new member. If you omit this clause, the description of the old member is copied.

Example

```
DUPLICATE MEMBER PAYAPP NEWNAME EMPAPP
DUP MEM PAYAPP NEW EMPAPP
```

DUPLICATE PACKAGE Command (DB2 Only)

The DUPLICATE PACKAGE command or equivalent prompter duplicates an existing package definition . To display the DUPLICATE prompter, select Option 5 on the Plan Menu or type DUPLICATE with no operands.

This command has the following format:

```
[ * ]
DUPLICATE [PACKAGE [name]] [TO] NEWNAME newname
```

Operand Definitions

Operand	Description
*	Duplicates the current package definition.
<i>name</i>	One- to seven-character name of a package definition.
<i>newname</i>	One- to seven-character name to give to the new package definition. This name must not be the name of an existing package definition in the current system.

DUPLICATE PLAN Command (DB2 Only)

The DUPLICATE PLAN command or equivalent prompter duplicates an existing plan definition. To display the DUPLICATE prompter, select Option 5 on the Plan Menu or type DUPLICATE.

This command has the following format:

```
          [ *          ]  
DUPLICATE [PLAN name] [TO] NEWNAME newname
```

Operand Definitions

Operand	Description
*	Duplicates the current plan DEFINITION.
<i>name</i>	One- to seven-character name of a plan definition.
<i>newname</i>	One- to seven-character name to give to the new plan definition. This name must not be the name of an existing plan definition in the current system.

EDIT DATAVIEW Command

The following command or equivalent prompter modifies an existing dataview definition for an unmodeled sequential or VSAM file dataview. This command displays the specified dataview definition for modification and makes it current.

Access the EDIT DATAVIEW prompter by selecting Option 1 on the Dataview Maintenance Menu (see the *Creating Dataviews Guide*) or issue the EDIT DATAVIEW command.

This command has the following format:

```
          [IDENTIFICATION]  
EDIT [* [FIELDS      ]]  
      [DATAVIEW dvw-name [VERSION nnn] [KEYS      ]]  
          [PARAMETERS  ]]
```

Operand Definitions

Operand	Description
*	You can substitute an asterisk for the current dataview name.
<i>dvw-name</i>	The 1- to 18-character dataview name of an unmodeled (sequential or VSAM file) dataview.
<i>nnn</i>	The specific version of the dataview to edit. If you do not specify a version, the version defaults to the version set in the SET VERSION or SET SITE VERSION command. For more information on specifying the version, see Using Version Clauses in CA Ideal Commands in the “Preliminary Concepts” chapter. nnn The one- to three-digit version number assigned to the dataview when it was created. LAST The latest version of the dataview definition, that is, the one with the highest version number.
IDENTIFICATION	Displays the dataview identification definition fill-in.
FIELDS	(Default) Displays the dataview field definition fill-in.
KEYS	Displays the key definition fill-in for a VSAM KSDS dataview.
PARAMETERS	Displays the dataview parameter definition fill-in.

EDIT JOBCARD Command

Each CA Ideal user is provided with a special fill-in called JOBCARD that you can edit to contain all JOB record information. Each site has a master JOBCARD fill-in. The jobcard member contains a maximum of five lines. CA Ideal automatically uses your jobcard member when you issue a PRINT command online with a destination of a system printer (resulting in a batch job submission).

This command has the following format:

```

                [MASTER      ]
EDIT JOBCARD [USER user-id ]

```

Operand Definitions

Operand	Description
MASTER	Displays the master JOBCARD for editing. If you omit this keyword, the user's JOBCARD displays.
<i>user-id</i>	One- to three-character ID of the user who owns the JOBCARD. This clause is required if the JOBCARD does not belong to the current user.

EDIT MEMBER Command

The EDIT MEMBER command displays a member. EDIT MEMBER alone produces a prompter.

This command has the following format:

```
[* [ {username}]]  
EDIT MEMBER [name [USER {user-id }]]
```

Operand Definitions

Operand	Description
*	You can substitute an asterisk for the current member name.
<i>name</i>	One- to eight-character member name.
<i>username</i>	The 1- to 15-character name of the user who owns the member. This clause is required only if the member does not belong to the current user.
<i>user-id</i>	One- to three-character ID of the user who owns the member. This clause is required if the member does not belong to the current user.

Example

```
EDIT MEMBER SIGNON
```


EDIT PACKAGE Command (DB2 Only)

The following command or equivalent prompter displays an existing package definition for editing. To display the EDIT prompter, select Option 1 on the Plan Menu or type EDIT.

This command has the following format:

```

      [*           ] [IDE]
EDIT [PACKAGE [name]] [RES]
                        [PAR]
  
```

Operand Definitions

Operand	Description
*	Displays the current package definition.
<i>name</i>	One- to seven-character name of a package definition.
IDE	Displays the package identification fill-in for editing.
RES	(Default) Displays the package resources fill-in for editing.
PAR	Displays the package parameters fill-in for editing.

EDIT PANEL Command

The following command or equivalent prompter displays an existing panel definition for editing and makes it current. When the panel definition is current, you can modify it.

You can edit another component of the current panel definition by issuing the name of one of the panel definition components, NEXT or PREVIOUS, without the EDIT command.

This command has the following format:

```

      { *           }
      {           {nnn} }
EDIT {PANEL name VERSION {PRODUCTION}} [component]
      {           {LAST} }
  
```

Operand Definitions

Operand	Description
*	Displays the current panel.
<i>name</i>	One- to eight-character name of the panel definition.
VERSION	The specific version of the panel. This entry is allowed (but not required) only if you specified a panel name. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the Preliminary Concepts” chapter.

Component	Description
IDENTIFICATION	Displays the panel identification definition fill-in.
PARAMETER	Displays the parameter definition fill-in.
LAYOUT	Displays the panel layout fill-in (default). You can specify LAYOUT BLANKFILL to edit the layout in blankfill mode or LAYOUT NULLFILL to edit the layout in nullfill mode. Blankfill is the default.
SUMMARY	Displays part of the layout in the top region of the screen and the field summary table in the bottom region of the screen. The fields in the top region are sequentially numbered. The numbers correspond to the field numbers displayed in the field summary table.
IRULES	Displays the field input editing/validation rules table.
ORULES	Displays the field output editing rules table.
FIELD <i>id</i>	Displays the field definition of the specified field. The field can be identified by name or number.
FACSIMILE	Displays the specified panel layout as it appears in its finished form.
NEXT	Displays the extended field definition for the field following the current field definition position.
PREVIOUS	Displays the extended field definition for the field that precedes the current field definition position.

EDIT PLAN Command (DB2 Only)

The following command or equivalent prompter displays an existing plan definition for editing. To display the EDIT prompter, select Option 1 on the Plan Menu or type EDIT.

This command has the following format:

```
[*          ] [IDE ]
EDIT [PLAN [name]] [RES ]
                               [DBRM]
                               [PAR ]
                               [PKL ]
```

Operand Definitions

Operand	Description
*	Displays the current plan.
<i>name</i>	One- to seven-character name of a plan definition.
IDE	Displays the plan identification fill-in for editing.
RES	(Default) Displays the plan resources fill-in for editing.
DBRM	Displays the plan DBRM fill-in for editing.
PAR	Displays the plan parameters fill-in for editing.
PKL	Displays the packlist fill-in for editing where packages are defined to the plan.

EDIT PROGRAM Command

The following EDIT command or equivalent EDIT prompter displays any component of an existing program definition and makes that definition the current definition. After the program definition is current, it is no longer necessary to name the program when accessing other components of the same program.

Access the Option 1 on the Program Maintenance Menu or by issuing the EDIT PROGRAM command.

This command has the following format:

```
[ *          ]
EDIT [PROGRAM name [VERSION version]] [ component ]
```

Operand Definitions

Operand	Description
<i>*</i>	Displays the current program definition. You can use the name of a program component (or equivalent function key) to locate another component in the current program definition.
<i>name</i>	One- to eight-character name of the program definition.
<i>version</i>	The specific version of the program. This entry is allowed (but not required) only if you specified a program name. If you omit both version and name, the default version of the current program displays. If you specify name and omit version, the default version of the named program displays. See the SET VERSION command for more information on setting the default version.

Operand	Description
<i>component</i>	<p>The component to edit. To display another component of the current program, you can enter the component name without specifying the EDIT command or press the equivalent function key. Valid components are:</p> <p>ENVIRONMENT (CA Datacom SQL only) Displays the program environment definition fill-in. Program environment data is defined and maintained using the fill-in shown in Creating Programs. You can specify default values for the fill-in using the SET DBSQL command.</p> <p>IDENTIFICATION Displays the program identification definition fill-in.</p> <p>RESOURCES Displays the program resources fill-in. The program resources fill-in specifies the resources the application uses. These resources can include dataviews, panels, reports, and subprograms.</p> <p>PARAMETER Displays the parameter definition fill-in. Parameter data consists of the names and descriptions of data items to pass to this program from a calling program or to this program using a RUN command.</p> <p>WORK Displays the working data definition fill-in.</p> <p>PROCEDURE (Default) Displays the procedure definition fill-in. If you enter this command before the procedure is defined, a blank screen appears, ready for PDL statements. If you enter this command after a procedure is defined, then as many lines of the procedure (from the top) as fit in the region display. The case of the text entered in the procedure fill-in is determined by the SET EDIT CASE command.</p> <p>Note: For non-deal programs, the only valid entries are IDENTIFICATION and PARAMETER.</p>

EDIT REPORT Command

The following EDIT command or equivalent EDIT prompter edits an existing report definition.

This command has the following format:

```
[ * ]
EDIT [REPORT name [VERSION version]] [component]
```

Operand Definitions

Operand	Description
*	Displays the current report definition.
<i>name</i>	One- to eight-character name of the specified report definition. If omitted, the default is the current report.
<i>version</i>	The specific version of the report. This entry is allowed (but not required) only if you specified a report name. If you do not specify version, the default version of the named report displays. See the SET VERSION command for more information on setting a default version.
<i>component</i>	The name of a report component to edit. To display another component of the current report definition, you can enter the component without specifying the EDIT command or press the equivalent function key. Valid components are: IDENTIFICATION Displays the report identification definition fill-in. PARAMETERS Displays the report parameters fill-in. HEADING Displays the report page heading definition fill-in. DETAIL (Default) Displays the report detail definition fill-in. COLUMN Displays the report column headings definition fill-in.

EDIT SYSTEM Command

The following EDIT command or equivalent EDIT prompter edits an existing system definition and makes it the current entity. See the *Administration Guide* for more information on creating and maintaining system definitions.

This command has the following format:

```

    { *                               }
    {      {sname}                     }
EDIT {SYSTEM {sid } [VERSION version] }
```

Operand Definitions

Operand	Description
*	You can substitute an asterisk for the name of the current system.
<i>sname</i>	The 1- to 15-character system name of the system definition. (The system name is the name of the system definition.)
<i>sid</i>	Three-character system short identifier of the system definition.
VERSION <i>version</i>	The specific version of the system. This entry is allowed (but not required) only if you specified a panel name.

If you do not specify version, the default version displays. For information on how to set the default version, see the SET VERSION command.

EDIT USER Command

The following EDIT command or equivalent EDIT prompter edits an existing user definition and makes it the current entity. For more information on creating and maintaining user definitions, see the *Administration Guide*.

This command has the following format:

```

      { *                               }
      { {username}                       }
EDIT {USER {user-id} [VERSION version] }

```

Operand Definitions

Operand	Description
*	You can substitute an asterisk for the name of the current user.
<i>username</i>	The 1- to 15-character user name of the user definition. (The user name is the name of the user definition.)
<i>user-id</i>	Three-character user short identifier of the user definition.

<i>version</i>	Specific version of the user definition. This entry is allowed (but not required) only if you specified a panel name. If you did not specify version, the default version of the named user definition displays. For more information on how to set the default version, see the SET VERSION command.
----------------	--

ENABLE RUN Command

The ENABLE RUN command lets you start running applications that were previously disabled.

If DISABLE and ENABLE commands are executed after a user left an active application, the run still aborts when the user continues the run. This is true even if the application is enabled before the session continues. This is because the user was running a program that is being replaced and the updateable portion of the program is from the old version of the program while the reentrant portion of the program is from the new copy. To synchronize all program components, the run must still be aborted and the user must restart the application.

This command has the following format:

ENABLE RUN *pgm-name*

Operand Definitions

Operand	Description
<i>pgm-name</i>	Specifies the highest-level program of an application that is in the “disabled” state. Execute a SELECT SYSTEM command before the ENABLE command to identify which CA Ideal system contains the disabled program. If an application was DISABLED in multiple online environments, then you must enable the application in each of those environments.

ENVIRONMENT Command (CA Datacom SQL Access)

This command displays the environment definition fill-in for the current program definition.

This command has the following format:

ENVIRONMENT

The environment definition fill-in sets the CA Datacom SQL access plan options for the program. If the site also has DB2, the fill-in also specifies the primary database, that is, the database management system SQL statements that can access only one database management system in a program. For more information, see the SQL concepts section in the *Programming Reference Guide*.

Program environment data is defined and maintained using the fill-in shown in the *Creating Programs Guide*. You can specify default values for the fill-in using the SET DBSQL command.

EXECUTE Command

The EXECUTE command initiates the processing of CA Ideal commands in a member. Storing commands in a member is useful when the same sequence of commands is used repetitively.

Each command stored in the member is successively executed until one of the following occurs:

- The last line is processed.
- A command causes an error. Subsequent commands in the member are not processed.
- Terminal I/O occurs. Subsequent commands in the member are not processed.

Note: CA Ideal does not continue processing each command stored in a member if one of the commands is a RUN that causes a program to transmit a panel. If a panel is transmitted, control is not passed back to the member and the commands following the RUN command do not execute.

You can continue execution after an error by using the SET ENVIRONMENT EXECERROR command in the member. However, SET ENVIRONMENT EXECERROR does not execute subsequent commands if there was a run time error. A run time error stops execution of the member even if you specified CONTINUE.

This command has the following format:

```
[ *  
[ [ {username}]]  
EXECUTE [member-name [USER {user-id }]]
```

Operand Definitions

Operand	Description
*	Executes the current member.
<i>member-name</i>	One- to eight-character member name.
<i>username</i> <i>user-id</i>	The 1- to 15-character user name or the one- to three-character user ID required. This clause is required only if the member belongs to another user.

Example

```
EXECUTE SIGNON
```

Re-executes the user's signon member.

GENERATE PACKAGE Command (Batch, DB2 Only)

The following command prepares an application to use static SQL to run and create a DB2 package. It is available only in batch.

GENERATE PACKAGE invokes the procedure that creates a DB2 application package. This procedure includes preparing the CA Ideal SQL module for the DB2 precompile and performing the DB2 precompile, assembly, link-edit, and bind.

For more information, see the *Administration Guide*.

This command has the following format:

```
GENERATE PACKAGE name [OWNER auth-id] [DESTINATION dest]
```

Operand Definitions

Operand	Description
<i>name</i>	One- to seven-character name of a package definition.
<i>auth-id</i>	One- to eight-character authorization ID of the owner of the package (DB2 Release 2 and higher).
<i>dest</i>	Destination of the Impact report produced during the generate process.
MAIL ' <i>email-id</i> '	A delimited 1- to 60-character name of a [assign the value for email in your book] destination.
LIBRARY	Output library.
SYSTEM <i>name</i>	System printer name.

GENERATE PLAN Command (Batch, DB2 Only)

The following command prepares an application to use static SQL to run and create a DB2 application plan. It is available only in batch.

GENERATE PLAN invokes the procedure that creates a DB2 application plan. This procedure includes preparing the CA Ideal SQL module for the DB2 precompile and performing the DB2 precompile, assembly, link-edit, and bind.

For further information, see the *Administration Guide*.

This command has the following format:

```
GENERATE PLAN name [OWNER auth-id] [DESTINATION dest]
```

Operand Definitions

Operand	Description
<i>name</i>	One- to seven-character name of a plan definition.
<i>auth-id</i>	One- to eight-character authorization ID of the owner of the plan (DB2 Release 2 and higher).
<i>dest</i>	Destination of the Impact report produced during the generate process.
MAIL ' <i>email-id</i> '	A delimited 1- to 60-character name of a [assign the value for email in your book] destination.
LIBRARY	Output library.
SYSTEM <i>name</i>	System printer name.

HELP Command

The HELP command or equivalent function key displays a panel or series of panels that explain how to complete the current panel or that describes the specified command or statement.

This command has the following format:

```
[command ]
[function ]
HELP [statement]
[COMMAND ]
[FUNCTION ]
[HELP    ]
[LINE    ]
[NEW     ]
[SAMPLE  ]
[SCROLL  ]
[SITE    ]
[STATEMENT]
[SYNTAX  ]
```

HELP without operands describes the current panel or fill-in. Press the PF2 (RETURN) key to return to the original activity.

When you specify an operand on the HELP command, a HELP member displays. The HELP session allows use of editor commands such as POSITION, FIND, and SCROLL.

Operand Definitions

Operand	Description
<i>command</i>	Any CA Ideal command that you can enter in the command area.
<i>function</i>	Any PDL function.
<i>statement</i>	Any PDL statement.
COMMAND	Displays a list of available commands. Commands are entered in the command area of the screen.
FUNCTION	Displays a list of available PDL functions.
HELP	Displays a list of available commands and statements.
LINE	Displays a list of available line commands. Line commands are entered in the margin area of the screen.
NEW	Describes the new features of the current release.

Operand	Description
SAMPLES	Provides a list of sample CA Ideal programs.
SCROLL	Provides an explanation of scrolling commands used in HELP members and throughout CA Ideal.
SITE	Provides instructions for adding or modifying CA Ideal help text.
STATEMENT	Displays a list of available PDL statements.
SYNTAX	Provides an explanation of notation used in all HELP members.

IDENTIFICATION Command

The following command or equivalent PF key displays the identification component of the current definition.

This command has the following format:

IDENTIFICATION

The program identification fill-in is described in the *Creating Programs Guide*. The panel identification fill-in is described in the *Creating Panel Definitions Guide*. For more information on the report identification fill-in, see the *Generating Reports Guide*.

IDENTIFY MODULE Command (Batch Only)

You must execute the IDENTIFY command in batch and only after a program was moved through an IBM utility. It makes the module known to CA Ideal at the receiving site. One IDENTIFY command is sufficient for all load modules associated with the specified program or panel.

Issue a SELECT SYSTEM command before the IDENTIFY command to indicate the CA Ideal system that contains the program or panel.

This command has the following format:

```
IDENTIFY MODULE modname [FOR] {PROGRAM pgm-name}  
                                {PANEL pnl-name }
```

Operand Definitions

Operand	Description
<i>modname</i>	One- to seven-character name specified in the CREATE MODULE command.
<i>pgm-name</i>	Name of the program that corresponds to this module.
<i>pnl-name</i>	Name of the panel that corresponds to this module.

IF ELSE ENDIF Commands

This construct is used in a CA Ideal batch jobstream to bypass execution of any other command based on the value of the return code.

The ELSE clause is optional. If the condition evaluates true, all commands are executed until the ELSE or ENDIF is encountered and all commands after ELSE are bypassed. If the condition is false, no commands are executed until either an ELSE or an ENDIF is encountered.

You cannot nest the IF-ELSE-ENDIF command series in other sets of IF-ELSE-ENDIF commands.

This command has the following format:

```

                                {GE}
                                {GT}
IF {$RETURN-CODE}             {LE} nnn
    {$RC          }           {LT}
                                {EQ}
                                {NE}

    CA Ideal commands
[ELSE                          ]
[  CA Ideal commands ]
ENDIF

```

Operand Definitions

Operand	Description
<i>nnn</i>	A numeric value tested against the current return code using the requested relational operator. The relational operator must be in the abbreviated format. You cannot omit it.

IRULES Command

The following CA Ideal command locates the field input editing/validation rules table for all the fields in a panel definition. You can modify information in this table. It is duplicated in the extended field definition.

This command has the following format:

```
IRULES
```

KEEP OUTPUT Command

The KEEP OUTPUT command is valid only during the DISPLAY OUTPUT activity. This command changes the output's disposition to KEEP and leaves the output in the output library after browsing. (When the output disposition is RELEASE, it is deleted upon termination of the browsing activity.)

To change the output's disposition to KEEP outside of DISPLAY OUTPUT activity, use the ALTER OUTPUT command.

This command has the following format:

```
KEEP [OUTPUT]
```

LAYOUT Command

This command is available only during PDF processing. It accesses the panel layout fill-in. The commands NULLFILL and BLANKFILL are available during layout editing to switch the fill mode. PF21 was assigned as a toggle between the modes.

The fill mode, either blankfill or nullfill, affects how the panel layout is edited.

Creating a new field on an empty line or after the last entry on the line:

- A null-filled screen requires leading characters to position the fields, otherwise, the fields shift left after you press the Enter key. To maintain space (a gap) between fields, enter spaces in place of the null characters. To leave a blank line, enter at least one space on the line.
- A blank-filled screen retains the fields in the exact position they are placed after you press the Enter key.

Inserting fields before an existing field:

- A null-filled screen inserts data by pressing the Insert key and then typing the data.
- A blank-filled screen requires you delete as many characters as you are inserting before inserting the new data.

Trailing blanks and trailing blank lines are not retained with the panel definition. They are only present during layout editing.

This command has the following format:

```

      {BLANKFILL}
LAYOUT {NULLFILL }

```

Operand Definitions

Operand	Description
BLANKFILL	Pads the lines in the panel layout with blanks to the full width of the panel.
NULLFILL	Does not pad the lines in the panel layout with blanks.

MARK STATUS Command

The MARK STATUS command or equivalent MARK STATUS prompter sets a version of a definition to production or history status. For further information on marking the status of a definition, see *Working in the Environment Guide*.

This command has the following format:

```

      { *
      { {DATAVIEW}
      { {PROGRAM }
      { {PANEL }
MARK [STATUS] { {REPORT } name VER version
      { {USER }
      { {SYSTEM }
      { {USER }

```

Operand Definitions

Operand	Description
*	An asterisk indicates the current entity occurrence.
<i>name</i>	Definition to mark: <ul style="list-style-type: none"> ■ 1- to 18-character name of an unmodeled dataview ■ 1- to 8-character program, report, or panel name ■ 1- to 15-character user name or system name ■ 1- to 3-character user-ID or system-ID
<i>version</i>	Version of the existing definition. For more information about specifying valid versions, see Using Version Clauses in CA Ideal Commands in the chapter “Preliminary Concepts.”
HIST	Specifies the new status for this version of the definition as the history-status version.
PROD	Specifies the new status for this version of the definition as the production-status version.

NULLFILL Command

This command is available only in panel definition layout editing. NULLFILL specifies that the panel layout is not padded with blanks. See the BLANKFILL command to edit the panel layout padded with blanks.

This command has the following format:

NULLFILL

Note: You can use PF21 to toggle between blankfill and nullfill mode.

OFF Command

This command signs off from a CA Ideal session. The sign-off panel appears when this command is used. The system supplies all information on this panel. It does not require user input. In split screen mode, OFF eliminates the last region.

This command has the following format:

OFF

OFFON Command

This command signs off from a CA Ideal session and requests a new CA Ideal sign-on panel.

This command has the following format:

```
OFFON
```

ORULES Command

This command locates the field output editing rules table for all of the fields in a panel definition. The table lists all fields appearing in the panel definition with a column for each of that field's output editing attributes. You can modify information in this table. It is duplicated in the extended field definition.

This command has the following format:

```
ORULES
```

PARAMETER Command

This command or its equivalent PF key locates the parameter definition for the current program definition. A parameter definition is allowed for a non-ideal subprogram.

Parameter data consists of the names and descriptions of data items to pass to this program from a calling program or to this program using a RUN command. Parameter data is specified for the called program on the parameter definition fill-in.

This command has the following format:

```
PARAMETER
```

PRINT Command

The PRINT command or equivalent PRINT prompter prints a specific definition. Access the PRINT prompter by selecting the PRINT option on the appropriate Maintenance Menu or by issuing a PRINT command.

Other PRINT commands are described under PRINT INDEX, PRINT INDEX MODULE, PRINT MEMBER, PRINT OUTPUT DESTINATION, PRINT OUTPUT STATUS, PRINT PLAN, PRINT SCREEN, and PRINT SESSION OPTIONS.

This command has the following format:

```

      { * }
      {{PROGRAM} }
      {{PANEL } }
PRINT {{REPORT } name [VERSION version] }
      {{SYSTEM } }
      {{USER } }
      {OUTPUT output }
      {DATAVIEW [auth-id]dvw-name [VERSION version] }

      [ {MAIL 'email-id' } ]
      [ {LIB } ]
      [DESTINATION {{SYS name} } ]
      [ {{NET name} COPIES nn } ]

      [NAME print-output]

      [ {KEEP } ]
      [DISPOSITION {RELEASE}]
      [ {HOLD } ]

      [DESCRIPTION 'string']
    
```

Operand Definitions

Operand	Description
*	You can substitute an asterisk for the entity type and entity name for the current entity.
<i>name</i>	Definition to print: <ul style="list-style-type: none"> ■ One- to eight-character program, report, or panel name ■ 1-to 15-character system or user name ■ One- to three-character system ID or user ID

Operand	Description
<i>version</i>	Version of the existing definition. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the “Preliminary Concepts” chapter.
output	One- to eight-character output name or the output number.
auth-id	One- to eight-character authorization ID required for SQL dataviews only.
dvw-name	The 1- to 18-character name of a dataview.
DESTINATION	Specifies the output destination. MAIL 'email-id' A delimited 1- to 60-character name of a [assign the value for email in your book]destination. LIBRARY Output library. SYSTEM name A system printer name. NETWORK name A network printer (not available in batch). Note: The same output printed at multiple network printers retains the NET DEST options of the first print request. For example, output number 5 printed at DEST NET PRT1, defined with HDR NO, is printed without a header at any DEST NET regardless of how the network printer is defined in CA Ideal.
COPIES <i>nn</i>	Number of copies to print on a system or network printer. This clause is ignored in batch.
NAME print-output	Name to assign to the output if the destination is to the output library.

Operand	Description
DISPOSITION	<p>Specifies a disposition. When a print request is issued (online or batch) and the destination is the output library, output is placed in the output library for browsing at the terminal.</p> <p>When a print request is printed online and the destination is a system or network printer, the output is placed in the output library with the disposition having the following effects:</p> <p>KEEP The job is printed and a copy of the output is retained in the output library.</p> <p>RELEASE The job is printed and a copy of the output is retained in the output library.</p> <p>HOLD The output is held until released.</p> <p>When a print request is issued in batch and the destination is a system printer, the output is printed on the system printer and no copy is retained in the output library.</p>
DESCRIPTION ' <i>string</i> '	A 1- to 32-character description of the output.

PRINT INDEX Command

The PRINT INDEX command lists the name and status of each occurrence of the specified entity type.

Optionally, the index can include occurrences of entity types that are related to a given entity type or occurrence (for example, each panel related to a given program). This display is based on dictionary facility relationships. Authorization in the specified system is required when relating dataviews or programs to systems. PRINT INDEX without an entity type calls up a prompter.

See also the PRINT INDEX OUTPUT command.

This command has the following format:

```
PRINT INDEX [ALL] [ent-type] [ent-name] [VERSION version]
           [USER user-name]
           [RELATED [TO] ent-type [ent-name] VERSION version]
           [           {MAIL 'email-id'           }]
           [           {LIB                       }]
           [DESTINATION {{SYS name}                }]
           [           {{NET name} COPIES nn       }]
           [DESCRIPTION 'string']
```

Operand Definitions

Operand	Description
ALL	<p>You can enter this parameter only when <i>ent-type</i> is PROGRAM. It prints an index of all programs, regardless of system, related to a specified program. When ALL is specified, the RELATED clause must be specified with a unique <i>ent-name</i>.</p> <p>Because the ALL option prints a cross-system program index, you must have an authorization of ADMIN to use this option.</p> <p>Note: The ALL operand requires that a version, where applicable, uniquely qualify the RELATED TO entity. It does not take a default.</p>
<i>ent-type</i>	<p>Classification of entities to index: that is, SYSTEM, USER, DATAVIEW, PROGRAM, PANEL, REPORT, MEMBER, PACKAGE, or PLAN. Printing the index of outputs is described under PRINT INDEX OUTPUT.</p> <p>PACKAGE and PLAN entities cannot be related to another entity type. In the RELATED TO clause, you cannot specify MEMBER, PACKAGE, and PLAN as an entity type.</p> <p>If <i>ent-type</i> in the RELATED clause is PROGRAM, the specified resources of each program are printed across all systems. For example, the command PRINT INDEX PNL RELATED TO PGM prints an index of all panels used by each program, regardless of the system that contains the panels.</p> <p>Note: If the <i>ent-type</i> is PROGRAM for both clauses and ALL is not specified, subprograms in other systems are listed for each program, but calling programs in other systems are not included.</p>

Operand	Description
<i>ent-name</i>	<p>(Optional) Specific name of an entity occurrence to index. SQL dataview names must include the authorization ID (for example, AUTHID1.PAY).</p> <p>You can use the following characters in an entity name to mask characters in the name during the index search.</p> <p>* Marks the position in the entity name of a single character to mask. For example, PAY*1 matches PAY01, PAY21, and so on.</p> <p>Note: When you use * in a string, you must know the length of the string to match. If the name being tested has more characters than the masked value, the name does not match. For example, neither PAY*1 nor PAY* matches PAY011.</p> <p>+ Marks the position in the entity name beyond which all characters are masked. For example, DISPLAY INDEX PROGRAM IDE+ matches all names starting with IDE of any length. This command locates both PROGRAM IDE1 and IDE11. However, if the * mask character were substituted for the + mask character, the program IDE1 is located, but not the program IDE11.</p> <p>You can use the mask characters in any position except the first position. The + mask character must be in the last position specified.</p> <p>When you use mask characters to search for systems or users, use the full entity name, not the short ID.</p>
<i>user-id</i>	<p>One- to three-character user-id required for an index of members belonging to another user.</p>
<i>version</i>	<p>Version of the entity occurrence. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the “Preliminary Concepts” chapter.</p> <p>Note: Unmodeled dataviews are all in DD test status, regardless of whether they are PROD or HIST to CA Ideal. For example, if you specify PRINT INDEX DVW VER T001, the resulting output contains both HIST and PROD status entries that are version 1.</p>

Operand	Description
DESTINATION <i>dest</i>	Specifies the output destination. MAIL 'email-id' A delimited 1- to 60-character name of a [assign the value for emailp in your book] destination. LIBRARY Output library. SYSTEM name System printer name. NETWORK name Network printer (not available in batch).
COPIES <i>nn</i>	Number of copies to print on a system or network printer. This option is ignored in batch.
DESCRIPTION ' <i>string</i> '	A 1- to 32-character description of the output.

Printing Program Relationships

If you relate program entity type to program entity type, for example,

```
PRINT INDEX PROGRAM MY-PROG RELATED TO PROGRAM
```

the index includes all subprograms called by the specified program (MY-PROG) and all programs that call the specified program. This applies to each program if more than one is indexed.

Examples: Program Index

The following illustrates the index of all programs in system DOC produced by the PRINT INDEX PGM command. Descriptions of each column follow.

```
=>
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+ >>>
IDEAL  DISPLAY OUTPUT          OUT INDEX (00357)          DISPLAY

===== T O P =====
PRINT INDEX PROGRAM          SYS: DOC    November 4, 1904    15:18:06

-----
Name      Ver S R-S Sys Lang Description          Created Updated
-----
DEMO1    001 T PRV DOC IDEAL produce demrpt1    11/04/04 11/04/04
DEMO2    001 T PRV DOC IDEAL produce mult dwv details 11/04/04 11/04/04
DEMO3    001 T PRV DOC IDEAL produce demrpt3-xsys res 11/04/04 11/04/04
DEMO4    001 T PRV DOC IDEAL produce demrpt4    11/04/03 11/04/03
DEMO5    001 T PRV DOC IDEAL list for reporting    11/04/04 11/04/04
DEMO6    003 T PRV DOC IDEAL produce dempgm6 fm dwv  11/04/04 11/04/04
===== B O T T O M =====
```

Field Definitions

Field	Description
Name	One- to eight-character entity name.
Ver	One- to three-digit number, QSAM or SAM identifier, or SQL label ('DBSQL' or 'DB2') CA Ideal assigned when this version of the entity was created.
S	Status version of the entity: that is, the test-status version (T), the production-status version (P), or the history-status version (H).
R-S	Run-status of the entity: PRV Private SHR Shared RES Resident
Lang	Language used to create this entity. IDEAL is the default. ASM or COBOL identifies Assembler or COBOL, respectively. PLI is identifies PL/I programs.
Description	A 1- to 32-character description of the entity.
Created	Date when the entity was created.
Updated	Date when the entity was last accessed in edit mode.

Examples: Related Index

The following illustrates the index of dataviews related to all programs in system DOC produced by the command:

PRINT INDEX DATAVIEW RELATED TO PROGRAM

```

=>
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+ >>>
IDEAL  DISPLAY OUTPUT          OUT INDEX (00359)                                DISPLAY

===== T O P =====
PRINT INDEX DATAVIEW                                November 4, 1904  15:19:28

-----
Name          Ver S U Type Description          Created Updated
-----
======> Related to PGM DEM01   Version 001 Status TEST
CUSTOMERS     001 P N   ORDER ENTRY DATAVIEW    01/30/02 10/05/04
======> Related to PGM DEM02   Version 001 Status TEST
CUSTOMERS     001 P N   ORDER ENTRY DATAVIEW    01/30/02 10/05/04
ORDERS        001 P Y   ORDER ENTRY DATAVIEW    01/30/02 09/02/04
======> Related to PGM DEM03   Version 001 Status TEST
CUSTOMERS     001 P N   ORDER ENTRY DATAVIEW    01/30/02 10/05/04
======> Related to PGM DEM04   Version 001 Status TEST
CUSTOMERS     001 P N   ORDER ENTRY DATAVIEW    01/30/02 10/05/04
======> Related to PGM DEM05   Version 001 Status TEST
CUSTOMERS     001 P N   ORDER ENTRY DATAVIEW    01/30/02 10/05/04

```

Field Definitions

Field	Description
Name	One- to eight-character entity name.
Ver	One- to three-digit number CA Ideal assigned when this version of the entity was created.
S	Status version of the entity; that is, the test-status version (T), the production-status version (P), or the history-status version (H).
U	Dataview record can be updated (Y), or cannot be updated (N).
Description	A 1- to 32-character description of the entity.
Created	Date when the entity was created.
Updated	Date when the entity was last accessed in edit mode.

PRINT INDEX MODULE Command

The PRINT INDEX MODULE command determines which application programs and panels were converted into module format. You can also specify an optional WITH VERIFICATION clause to verify that the modules are really in the module library and represent the appropriate application program.

This command has the following format:

```
PRINT INDEX [ALL] MODULE [ [WITH] {VERIFICATION} ]
```

Operand Definitions

Operand	Description
PRINT INDEX	Prints an index of all the programs and panels in the “current system” that were converted to module format, based on the MODULE entities in Datadictionary. The module name, the corresponding application program or panel identification, and the date/time of program compilation display. The entries are in alphabetical order by module name, with all program modules first, followed by panel modules.
ALL	Gives an index listing of all programs and panels in all systems that were converted to module format. The order is alphabetical by module name. If you omit ALL, only modules for the current system display or print.
WITH VERIFICATION	Attempts to load each module or set of modules to determine whether the correct modules are available to the environment. The entity type, system ID, entity name, version number, and date/time stamp in the MODULE Datadictionary entity are compared with the information found in the loaded module. Any discrepancies are noted.

PRINT INDEX OUTPUT Command

The PRINT INDEX OUTPUT command lists the name and status of each output member for the current user. It is equivalent to PRINT OUTPUT STATUS.

This command has the following format:

```
PRINT INDEX OUTPUT
```

PRINT INDEX SESSION Command

The PRINT INDEX SESSION command lists the name and status of each current user. It provides information to determine who is active in the Session Control Facility (SCF) environment. CA products that run in the SCF environment are CA Ideal, Datadictionary Online, and CA MetaCOBOL+ Panel Definition Facility. You can only issue this command in the VSE and z/OS CICS environments.

Note: If you initiate an asynchronous task (a compile or network print) and log off before the completion message is sent, you still show up as logged on.

This command has the following format:

```
PRINT INDEX SESSION
```

The following example illustrates the index of users this command produces. Each column is described following.

Command	User	OPID	Reg	User name	Term	Product	Sys	Main PGM	Ver	Sub
000001	RAY	RAY	1	RAY	Z000	IDEAL	WOR	WORUN	PRD	0017
000002	MDF	MDF	1	MDUFFY	Z002	IDEAL	WOR	WORUN	PRD	0018
000003	SFT	SFT	1	SEIFERT	Z003	IDEAL		**NONE**		0000

B O T T O M

Field Definitions

Field	Description
User	One- to three-character CA Ideal user ID.
OPID	One- to three-character CICS op-ID for the user. This is the ID from the TCTTEOI (operator ID) field of the CICS TCTTE control block.
Reg	Terminal display region.
User name	The 1- to 15-character CA Ideal user name.
Term	Terminal ID.
Product	Name of the product used.
Sys	System where the main program resides, if a program is run.

Field	Description
Main PGM	Name of a program executed using a RUN command. If the first region is in CA Ideal but not executing a program, **NONE** is returned.
Ver	Version of the main program, if a program is run.
Sub	Number of programs in the run unit for the Main Program. The total is all of the programs defined in the resource section of all the programs that were called. If a program is released, the storage is released but the program is still counted in the total.

CICS Considerations

The command returns a snapshot of the environment. The environment can change before the information is returned. The information returned is retrieved from control blocks kept in CICS Auxiliary Temporary Storage records, which are updated at a transaction boundary. This implies that the information is not current if the user is an active CICS task.

For each user the CICS ID, the short three-character user ID, the user's full name, the terminal ID, and the name of the product logged onto is returned. If the user's terminal is split into multiple regions, a set of information is returned for each region. If a user is running a program in the CA Ideal environment (which can only occur in region 1), the main program, system, version, and the number of programs in the run unit are returned. You can issue the RUN command only in region one.

The IDs returned are the OP IDs defined for the user in the CICS signon table (SNT). If the signon table is not used, it is the ID placed in the TCTTEOI field of the terminal control table (TCT). All user IDs are returned regardless of the products the user is logged onto.

PRINT MEMBER Command

The PRINT MEMBER command generates an output of a member.

This command has the following format:

```

[ {username}]
PRINT MEMBER name [USER {user-id }]
[ {MAIL 'email-id' } ]
[ {LIB } ]
[DESTINATION {{SYS name} } ]
[ {{NET name} COPIES nn } ]
[DESCRIPTION 'string' ]
    
```

Operand Definitions

Operand	Description
<i>name</i>	One- to eight-character member name.
<i>username</i> <i>user-id</i>	The 1- to 15-character CA Ideal user name or the one- to three-character user ID of the user who owns the member. This clause is required if the member does not belong to the current user.
DESTINATION	Specifies the output destination. MAIL 'email-id' A delimited 1- to 60-character name of a [assign the value for email in your book] destination. LIBRARY Output library. SYSTEM name System printer name. NETWORK name Network printer (not available in batch).
COPIES <i>nn</i>	Number of copies to print on a system or network printer. This option is ignored in batch.
DESCRIPTION ' <i>string</i> '	A 1- to 32-character description of the member. For more information about description of valid string delimiters, see the chapter "Preliminary Concepts."

Example

```
PRINT MEMBER SIGNON DEST SYS LPT3
```

PRINT OUTPUT DESTINATION Command

This command generates an output member containing all valid output destinations.

This command has the following format:

```
PRINT OUTPUT DESTINATION[S]
```

The printed output looks just like the displayed output from the DISPLAY OUTPUT DESTINATION command.

PRINT OUTPUT STATUS Command

The PRINT OUTPUT STATUS command prints the status of a print request.

This command has the following format:

```

    {OWN  }
PRINT OUTPUT {ALL  } STATUS
    {name  }
    {number}

[      {MAIL 'email-id'      }]
[      {LIB                   }]
[DESTINATION {{SYS name}      }]
[      {{NET name} COPIES nn  }]
[DESCRIPTION 'string' ]
    
```

Operand Definitions

Operand	Description
OWN	Prints the status of output under the user's identification.
ALL	Prints the status of all output.
<i>name</i>	Prints the status of the output with the specified one- to eight-character name. The name must be unique; otherwise a message and a list of all outputs with that name appear.
<i>number</i>	Prints the status of the output with the specified 1- to 4-digit number.
DESTINATION	Specifies the output destination. MAIL 'email-id' A delimited 1- to 60-character name of a [assign the value for emailp in your book] destination. LIBRARY Output library. SYSTEM name System printer name. NETWORK name Network printer (not available in batch).
COPIES <i>nn</i>	Number of copies to print on a system or network printer only. This option is ignored in batch.
DESCRIPTION ' <i>string</i> '	A 1- to 32-character description of the output.

The format of the information the PRINT OUTPUT STATUS command produces is identical to the DISPLAY OUTPUT STATUS command.

Examples

The following illustrates the use of PRINT OUTPUT.

```
PRINT OUTPUT X STATUS DESTINATION SYSTEM LPT1
PRINT OUTPUT ALL STATUS DEST SYS LPT1
PRINT OUTPUT STATUS
```

PRINT PACKAGE Command (DB2 Only)

The following command or equivalent prompter prints an existing package definition. Preparing DB2 packages is described in the *Administration Guide*.

To display the PRINT prompter, select option 3 on the Plan Menu or type PRINT.

This command has the following format:

```
[ * ]
PRINT [PACKAGE [name] ] [[WITH] VERIFICATION]
[           {MAIL 'email-id'           } ]
[           {LIB                       } ]
[DESTINATION {{SYS dest-id}           } ]
[           {{NET dest-id} COPIES nn   } ]
```

Operand Definitions

Operand	Description
*	Prints the current package definition.
<i>name</i>	One- to seven-character name of a package definition.
WITH VERIFICATION	This clause collects data on the generate package process for the specified package without updating the DB2 plan tables or performing the DB2 BIND command. It automatically prepares the programs in the specified package and creates the BIND command that bind the package to the application and produces an impact report. Use the GENERATE PACKAGE command to prepare the program and perform the bind. VER is the minimum abbreviation.

Operand	Description
DESTINATION	Specifies the output destination. MAIL 'email-id' A delimited 1- to 60-character name of a [assign the value for email in your book] destination. LIBRARY Output library. SYSTEM name System printer name. NETWORK name Network printer (not available in batch).
COPIES <i>nn</i>	Number of copies to print on a system or network printer, online only. This option is ignored in batch.

PRINT PLAN Command (DB2 Only)

The following command or equivalent prompter prints an existing plan definition. Preparing DB2 application plans is described in the *Administration Guide*.

To display the PRINT prompter, select Option 3 on the Plan Menu or type PRINT.

This command has the following format:

```
[ *          ]
PRINT [ PLAN [name] ] [[WITH] VERIFICATION]
      [          {MAIL 'email-id'           } ]
      [          {LIB                       } ]
      [DESTINATION {{SYS dest-id}           } ]
      [          {{NET dest-id} COPIES nn    } ]
```

Operand Definitions

Operand	Description
*	Prints the current plan.
<i>name</i>	One- to seven-character name of a plan definition.
WITH VERIFICATION	This clause collects data on the generate plan process for the specified plan without updating the DB2 plan tables or performing the DB2 BIND command. It automatically prepares the programs in the specified plan and creates the BIND command that bind the plan to the application and produces an impact report. Use the GENERATE PLAN command to prepare the program and perform the bind. VER is the minimum abbreviation.

Operand	Description
DESTINATION	Specifies the output destination. MAIL 'email-id' A delimited 1- to 60-character name of a [assign the value for emailp in your book] destination. LIBRARY Output library. SYSTEM name System printer name. NETWORK name Network printer (not available in batch).
COPIES <i>nn</i>	Number of copies to print on a system or network printer, online only. This clause is ignored in batch.

PRINT SCREEN Command

This command generates an output of the current screen contents.

This command has the following format:

```

PRINT SCREEN [
    [ {MAIL 'email-id' } ]
    [DESTINATION {LIB } ]
    [ {{SYS dest-id} } ]
    [ {{NET dest-id} COPIES nn } ]
    [DESCRIPTION 'string' ]

```

Operand Definitions

Operand	Description
DESTINATION	Specifies the output destination. MAIL 'email-id' A delimited 1- to 60-character name of a [assign the value for emailp in your book] destination. LIBRARY Output library. SYSTEM name System printer name. NETWORK name Network printer (not available in batch).
COPIES <i>nn</i>	Number of copies to print on a system or network printer. This option is ignored in batch.
DESCRIPTION ' <i>string</i> '	A 1- to 32-character description of the output.

PRINT SESSION OPTIONS Command

This command prints the current value of every option or a specified type of option. The options shown include those set in the signon procedure and during the current session and the site defaults. For a list of options, see the DISPLAY SESSION OPTIONS command.

This command has the following format:

```
PRINT SESSION OPTION[S]
  [      {MAIL 'email-id'      } ]
  [DESTINATION {LIB          } ]
  [      {{SYS dest-id}      } ]
  [      {{SET dest-id} COPIES nn } ]
[DESCRIPTION 'string']
```

Operand Definitions

Operand	Description
DESTINATION	Specifies the output destination. MAIL 'email-id' A delimited 1- to 60-character name of a [assign the value for emailp in your book] destination. LIBRARY Output library. SYSTEM name System printer name. NETWORK name Network printer (not available in batch).
COPIES <i>nn</i>	Number of copies to print on a system or network printer. This option is ignored in batch.
DESCRIPTION ' <i>string</i> '	A 1- to 32-character description of the output.

PROCEDURE Command

The following command or equivalent PF key displays the program procedure for the current program definition.

If you enter this command before the procedure is defined, a blank screen appears, ready for PDL statements. If you enter this command after a procedure is defined, then as many lines of the procedure (from the top) as fit in the region display.

The SET EDIT CASE command case determines the text entered in the procedure fill-in.

This command has the following format:

```
PROCEDURE
```

PROCESS Command

This command displays the Process Program Menu.

This command has the following format:

```
PROCESS
```

PRODUCE REPORT Command

You create a report facsimile by using the PRODUCE REPORT command. Entering PRODUCE alone on the command line displays a prompter screen. CA Ideal requires a completed report definition and a compiled program containing the report in its resource section. (See Other Program Requirements following the syntax explanation additional requirements.) The PRODUCE REPORT command produces a skeleton report using arbitrarily generated data. The data is randomly generated. The Detail option limits the number of records produced.

Note: The PRODUCE REPORT command is valid for TEST status programs only.

This command has the following format:

```

      {*                }
PRODUCE {REPORT report-name}

      {*                }
[FOR] {PROGRAM pgm-name [VERSION version]} [DETAIL nnnn]

      [          {MAIL 'email-id'          }]
      [          {LIB                          }]
[DESTINATION {{SYS dest-id                }]
      [          {{NET dest-id} COPIES nn }]

      [          {KEEP   }]
[DISPOSITION {RELEASE}]
      [          {HOLD   }]

[MAXLINES n]

[DESCRIPTION 'string']

```

Operand Definitions

Operand	Description
*	Current program or report name.
<i>report-name</i>	Name of the report to produce.
<i>pgm-name</i>	Name of the program the PRODUCE REPORT command accesses.
Operand	Description
<i>version</i>	Version of the entity occurrence. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the "Preliminary Concepts" chapter.
DETAIL <i>nnnn</i>	Number of test detail records up to 9999.
DESTINATION	Specifies the output destination. MAIL 'email-id' A delimited 1- to 60-character name of a [assign the value for emailp in your book] destination. LIBRARY Output library. SYSTEM name System printer name. NETWORK name Network printer (not available in batch).
COPIES <i>nn</i>	Number of copies to print on a system or network printer. This option is ignored in batch.
DISPOSITION	The disposition clause; select from the following: KEEP RELEASE HOLD For more information, see SET OUTPUT Command.
MAXLINES <i>n</i>	Maximum number of lines for the report facsimile. The upper limit for MAXLINES is established at CA Ideal installation or by a SET OUTPUT SITE OPTIONS fill-in. It only applies to reports going to the output library. Any report reaching this maximum stops the run. This option does not affect reports produced and printed in batch.
DESCRIPTION ' <i>string</i> '	A 1- to 32-character description.

Other Program Requirements

You must define all fields defined in the report in the specified program's working data or parameter data or in a dataview that is included in the program's resource section and in a FOR construct executed by the program.

The facsimile uses only the report's primary group. The primary group must contain at least one data item (it cannot consist of literals without data).

Example

```
PRODUCE RPT ABC FOR PGM XYZ DEST LIBRARY
```

PROGRAM Command

This command displays the Program Maintenance Menu.

This command has the following format:

```
PROGRAM
```

PURGE Command

The PURGE command cleans up storage areas left allocated for a CA Ideal session that is abruptly ended in CICS. For example, if a line or a node is dropped and CICS autoinstalled the terminal, CAIVPE (Virtual Processing Environment) might have storage areas and control blocks still allocated on behalf of the terminal. If the session is not purged, these storage areas could be allocated indefinitely since the random terminal-ID assignment of AUTOINSTALL might not reuse the original terminal ID.

For more information on alternate ways to purge a CA Ideal session, see the *Administration Guide*.

This command has the following format:

```
PURGE term-id
```

Operand Definitions

Operand	Description
<i>term-id</i>	Four-character terminal ID. You must type this value exactly as generated by the AUTOINSTALL routine since term-ID is case sensitive. When the session is purged, the following message displays: Terminal ' <i>term-id</i> ' purge complete If the value supplied for term-ID is not a defined terminal ID, the following message displays: Terminal ' <i>term-id</i> ' not found Check to be sure you entered the terminal ID in the correct case, exactly as the AUTOINSTALL routine generated it.

REBIND Command (CA Datacom SQL Access)

This command requests that CA Datacom/DB rebind an access plan for the specified program. Before rebinding, you can modify the plan options in the program object code without recompiling the program using the ALTER PROGRAM ENVIRONMENT command. The REBIND command affects only the plan identified with the specified (or implied default) authorization ID, version number, and program name.

If you issue an ALTER PGM ENV command against a program associated with more than one plan, you must issue the REBIND command for each plan that requires the changed options.

If the specified program was converted to load module format, the REBIND command uses the load module. To modify the program's plan options without recompiling, use an ALTER PROGRAM ENVIRONMENT command with the program, then issue a CREATE MODULE.

This command has the following format:

```
          { *                               }  
REBIND {PROGRAM program-name [VERSION version]} [FOR AUTH authid]
```

Operand Definitions

Operand	Description
*	Represents the current program name.

<i>program-name</i>	One- to eight-character program name.
<i>version</i>	Version of the program. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the “Preliminary Concepts” chapter.
FOR AUTH <i>authid</i>	Identifies the specific plan to rebind, if more than one plan exists for the specified program. If you omit this clause, the authorization ID of the default plan (stored in the program object) is used.

REFORMAT Command

The REFORMAT command changes the distribution of the current number of lines in each region affected by the reformat (regions 1, 2, and possibly, 3).

This command has the following format:

```
REFORMAT n [m]
```

Operand Definitions

Operand	Description
<i>n</i>	Number of lines for the first region.
<i>m</i>	Number of lines for the second region.
	The remaining lines are occupied by the third region.

Example

For two regions, one amount is specified because the remainder of the screen appears for the second region. For example, the following command formats 15 lines in the first region and the remainder of the screen for the second region.

```
REFORMAT 15
```

For three regions, the first two amounts are specified and the third region occupies the remainder of the screen. For example, the following command formats 10 lines in each of the first and second regions and the third region occupies the remainder of the screen.

```
REFORMAT 10 10
```

Note: If the number of lines specified is greater than the number of lines available or less than the minimum lines per region, reformatting is ignored and a message is issued.

REFRESH Command

The REFRESH command makes a new copy of a CA Ideal program or panel available in load module format or makes a regenerated CA Ideal DB2 plan available.

Important! This command is for CICS only.

Programs and Panels

For programs and panels in load module format, the command is executed for each program or panel that was replaced or transported to a new system. You should also execute it in each environment using the replaced program or panel.

The command has two purposes.

- It updates the in-core module table that defines the application programs and panels that are in load module format (in CICS, PPT entries are automatically added if none exist) and verifies them.
- It makes the new modules available to the users.

To identify the CA Ideal system that contains the program or panel, execute a SELECT SYSTEM command before the REFRESH command. For more information on load modules, see the *Administration Guide*.

Plans

Use the REFRESH PLAN command when a DB2 application plan was regenerated. It makes the CA Ideal SQL modules for the plan available to the teleprocessing monitor, validates the date/time stamps of the SQL modules, and reloads a fresh copy of the module. See the *Administration Guide* for more information on plans.

This command has the following format:

```
{PLAN plan-name }  
  
{PACKAGE pkg-name}  
REFRESH {PROGRAM pgm-name}  
{PANEL pnl-name }
```

Operand Definitions

Operand	Description
<i>pkg-name</i>	DB2 package to reload.

<i>plan-name</i>	DB2 plan to reload.
<i>pgm-name</i>	Program to reload.
<i>pnl-name</i>	Panel to reload.

Other Program Requirements

Run this command only online.

Before issuing this command, issue a DISABLE RUN command for any applications that use the programs, panels, packages or plans being refreshed. Make sure that all access to these components is stopped.

REPORT Command

The functions CA Ideal offers to define and maintain report definitions are presented in the Report Maintenance Menu. To access this menu, select the appropriate option on the Main Menu or enter the equivalent CA Ideal command.

This command has the following format:

REPORT

RESET Command

This command resets the assignments made by ASSIGN commands. It returns the assignments of options for authorization IDs, reports, dataviews, DBIDs, or programs to the assignment made before the session started or to that provided in the RUN command.

During an edit session, use RESET to cancel any pending line command operations.

During panel layout editing of the Panel Definition Facility, the pending copy and move operations and any designated destinations are canceled.

This command has the following format:

```
[AUTHORIZATION auth-id]
[REPORT name      ]
RESET [PROGRAM name ]
[DATAVIEW name   ]
[DBID nnn        ]
```

```
[AUTHORIZATION]
[REPORT        ]
RESET ALL [PROGRAM ]
[DATAVIEW     ]
[DBID         ]
```

Operand Definitions

Operand	Description
RESET	<p>With no operands, cancels any pending functions. During most editing, RESET cancels pending line commands. During PDF layout editing, RESET cancels pending copy and move functions by replacing the copy and move symbols with the start-field symbol and removing any destination symbols from the layout.</p> <p>With an entity type and name, resets the specific authorization ID, report, program, dataview, or DBID.</p>
RESET ALL	<p>With no other operands, resets all authorization IDs, dataviews, DBIDs, programs, and reports that are currently assigned.</p> <p>With an entity type, resets all of the specified entities that are currently assigned. For example, RESET REPORT resets all currently assigned reports.</p>

RESOURCES Command

The following command or equivalent PF key displays the program resources component of the current program definition. The program resources fill-in shown in the *Creating Programs Guide* specifies the resources the application uses. These resources can include dataviews, panels, reports, and subprograms.

This command has the following format:

```
RESOURCES
```

RUN Command

Execution of a program occurs when you enter a complete RUN command or when a prompter equivalent to that command is transmitted. The RUN command specified with no operands displays the RUN prompter. RUN * executes the current program.

The UPDATE parameter determines whether or not updates will be executed or suppressed for all dataviews and tables (for example, Datacom/CBS, Datacom/SQL, DB2, Sequential, and VSAM). This parameter refers to data accessed via the FOR construct, by embedded SQL, or both. The parameter can be useful for debugging or testing a program without actually updating the data.

The UPDATE parameter is not intended to provide security, nor is it intended as an easy way to disable the update feature of a program. Programs will function as if the update was successful. There will be no indication that the record or records were not successfully updated. If the program supplies messages which indicate that an add or update was successful, these messages will still be produced. However, the adds and updates will not have really taken place.

The record will only reflect the requested change until another record is accessed within the scope of the FOR construct or SQL statement. The parameter only suppresses the internal I/O commands that perform the addition, update, or deletion of the given record. All other processes are carried out. For example, a sequential file will still be allocated and the DISP parameter in the JCL stream will still determine whether or not the file is retained when the execution completes.

Suggested uses for the UPDATE parameter:

- Distinguish update programs from inquiry only programs.
- Adjust the code to be sensitive to update requests based on PF key selection or input parameters.
- Create updateable versus non-updateable panels that can be used alternatively under various conditions.

This command has the following format:

```

{ *                               }
RUN {pgm-name [VERSION version] }

[                               [Y]]
[ UPDATE [DB][N]]

[PARAMETER] 'string'

[MCPARM ]

[ {MAIL 'email-id' } ]
[ {LIB } ]
[DESTINATION {{SYS dest-id} } ]
[ {{NET dest-id} COPIES nn}]

[ {KEEP } ]
[DISPOSITION {RELEASE}]
[ {HOLD } ]

[LOOPLIMIT nnnnnn]

[MAXLINES n]

[DESCRIPTION 'string']
    
```

Operand Definitions

Operand	Description
<i>pgm-name</i>	Name of the program executed.
<i>version</i>	Version of the program. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the “Preliminary Concepts” chapter.
UPDATE	Either updates or bypasses updating the database. Y Allows updating of the database. N Does not allow updating of the database.
PARAMETER ' <i>string</i> '	A single string using uppercase translation that is passed to the program. For a description of a valid string delimiter, see Basic Steps to Create a Report earlier in this guide.

Operand	Description
MCPARM ' <i>string</i> '	<p>Single string using mixed case translation that is passed to the program. For a description of a valid string delimiter, see Basic Steps to Create a Report earlier in this guide.</p> <p>You must define this parameter in the parameter definition fill-in for the program being run. The length of the string is limited to the space available in the prompter. When you issue the RUN command in the command area, the command with parameter string is limited to one command line.</p> <p>You must define this parameter in the parameter definition fill-in for the program run as one alphanumeric (type X) character string with input Update Intent (not a group item).</p>
DESTINATION	<p>This destination information overrides the SET OUTPUT DESTINATION session option. It provides a default destination for RUNLIST and for any reports produced in this run. You can override this destination with the ASSIGN REPORT command before this RUN command or with the ASSIGN REPORT statement during this run.</p> <p>This clause differs from the more typical DESTINATION-clause in that "NAME <i>name</i>" is excluded. The one- to eight-character report name becomes the output name. Select from the following:</p> <p>MAIL 'email-id' A delimited 1- to 60-character name of [assign the value for emailp in your book] destination.</p> <p>LIBRARY Output library.</p> <p>SYSTEM name System printer name.</p> <p>NETWORK name Network printer (not available in batch).</p>
COPIES <i>nn</i>	<p>Number of copies to printed on a system or network printer. This option is ignored in batch.</p>
DISPOSITION	<p>Disposition clause; select from the following:</p> <p>KEEP</p> <p>RELEASE</p> <p>HOLD</p> <p>For more information about these clauses, see the SET OUTPUT Command.</p>
LOOPLIMIT <i>nnnnn</i>	<p>Maximum number of times any loop in the program may be consecutively executed. The count is reset each time the loop is entered. Unlike the value set by SET RUN LOOPLIMIT, this operand applies to PROD runs.</p>

Operand	Description
MAXLINES <i>n</i>	Maximum number of lines for any one report that a program can produce. The upper limit for MAXLINES is established at installation or by a SET OUTPUT MAXLINES command. It only applies to reports going to the output library. Any report reaching this maximum stops the run. This option does not affect reports produced and printed in batch.
DESCRIPTION ' <i>string</i> '	A 1- to 32-character description of the list output (RUNLIST). The description displays as part of the output status (see also the DISPLAY OUTPUT STATUS command in this guide). The default description LIST STATEMENT OUTPUT applies if you do not provide a description in the RUN command. You can override the description before the run using the ASSIGN REPORT command or during the run using the ASSIGN REPORT statement.

SELECT SYSTEM Command

A system is a collection of application programs and the developers and users associated with it. A user can only display, edit, run, and so on, programs, panels, and reports in the current system. A system must be assigned to a user before he can select that system. (Defining a user and assigning systems is described in the *Administration Guide*.)

If the user is associated with only one system, that system becomes the active or “current system” upon sign on. If the user is authorized to use more than one system, the first system (alphabetically) is selected as the current system at sign on. To select another system as the current system at any time during the session, issue the following command.

Note: You can also include the SELECT SYSTEM command as part of a user's signon member.

This command has the following format:

```
                {sname}  
SELECT SYSTEM {sid }
```


Operand Definitions

Operand	Description
<i>sname</i>	The 1- to 15-character system name of the system definition. The system name is the name of the system definition.
<i>sid</i>	Three-character system short identifier of the system definition.

SET \$RETURNCODE Command

You can use the following command in the jobstream to control the value of the return code.

This command has the following format:

```
{ $RC      }
SET { $RETURN-CODE } EQ nnnn
```

Operand Definitions

Operand	Description
<i>nnnn</i>	Value for the return code in the range of 0 - 9999.

SET ASYNCMMSG Command

This command determines whether to display asynchronous messages on the terminal. This includes all messages indicating the completion of prints and compiles for the user.

This command has the following format:

```
{ USER }
SET [ SITE ] ASYNCMMSG { NONE }
```

Operand Definitions

Operand	Description
USER	Displays all asynchronous messages whose short IDs match the OPID or Security ID.

NONE	Does not display asynchronous messages.
------	---

SET CATALOG VALIDATION Command

This command defines the session or site default for validation of VSAM dataviews when they are cataloged. If a default is not set with this command and no validation options are specified on the CATALOG command, the dataview is not validated when it is cataloged.

This command has the following format:

```
                                {YES}  
SET [SITE] CATALOG VALIDATION {NO}
```

Operand Definitions

Operand	Description
YES	Specify this option to define the default option of the CATALOG command as WITH VALIDATION. This means that, by default, VSAM dataviews are validated when the dataview is cataloged.
NO	Specify this option to define the default option of the CATALOG command as WITH NOVALIDATION. This is the installed default and means that, by default, VSAM dataviews are not validated when cataloged.

SET SITE CHECK DUPLICATE Command

For CICS only, this command determines whether CA Ideal ensures that all users are signed on to CA Ideal with a unique CICS user (OPID or external security ID).

For more information on SCBOPTCB, see the *CA IPC Customization and Tuning Guide*. For more information on sign-on security, see the *Administration Guide*.

This command has the following format:

```
                                {YES}  
SET SITE CHECK DUPLICATE USER {NO }
```

Operand Definitions

Operand	Description
YES	Ensure unique CICS user. Note: Yes is only effective within a single CICS region. YES is not recommended for a MRO environment with multiple Application Owning Regions. External security software can be used to ensure unique CICS users.
NO	Allow non-unique CICS user.

SET COMMAND Command

This command sets individual command options or displays a fill-in screen that sets multiple command options for the session or for the site.

This command has the following format:

```

      { COMMENT x           }
      {                     }
      {           { A }     }
      { DATEFOR { E }       }
      {           { I }     }
      { DELIMITER x        }
      { LINE n             }
      {                     }
      { REPEAT x           }
      {                     }
      { RESHOW x           }
      {                     }
SET COMMAND { RESHOW { ON } }
            { OFF }       }
      {                     }
      {           { x }     }
      { SEPARATOR { N }     }
      {           { G }     }
      {                     }
      { SESSION [OPTIONS]  }
      {                     }
      { SITE [OPTIONS]     }
      {                     }
      { UPPERCASE { YES }   }
      {           { NO }   }

```

Operand Definitions

Operand	Description
COMMENT	<p>Defines the character that sets off a comment from a command in the command area. Specify:</p> <p>x Any character with the exception of an underscore (<code>_</code>) or the letter <code>S</code>.</p>
DATEFOR	<p>Establishes a default date format. Specify:</p> <p>A American (mm/dd/yy) E European (dd/mm/yy) I International (yy/mm/dd)</p>
DELIMITER	<p>Defines the character that separates commands entered on the same command line. Specify:</p> <p>x Any character with the exception of an underscore (<code>_</code>) or the letter <code>S</code>.</p> <p>Note: The command delimiter is valid only for screen input online.</p>
LINE	<p>Defines the number of lines in the command area. Specify:</p> <p>n From zero to five lines.</p>
REPEAT	<p>Defines the character that reexecutes the previous command. Specify:</p> <p>x Any character with the exception of an underscore (<code>_</code>) or the letter <code>S</code>.</p>
RESHOW	<p>Defines the character that redisplay the previous command. Specify:</p> <p>x Any character with the exception of an underscore (<code>_</code>) or the letter <code>S</code>.</p>
RESHOW	<p>Defines the command reshow status. Specify:</p> <p>ON Redisplays all subsequent commands after execution. OFF Disables the display of the last executed command (except when in error).</p>
SEPARATOR	<p>Defines the character that separates the command area from the display area. Specify:</p> <p>x Any character. The separator line consists of the specified character.</p> <p>N Null. The separator line is suppressed.</p> <p>G Grid. The separator line is a scalar grid.</p>

Operand	Description
SESSION [OPTIONS]	Calls up a fill-in in which you can set the command area options for the current session. For more information on complete explanation of the fill-in, see the <i>Working in the Environment Guide</i> .
SITE [OPTIONS]	Calls up a fill-in in which you can set the command area options for the site. For more information on complete explanation of the fill-in, see the <i>Administration Guide</i> .
UPPERCASE	Determines whether CA Ideal displays system panels and messages in uppercase and can be abbreviated as SET CMD UPC. Specify: YES Displays system messages and panels in uppercase. NO Displays system messages and panels in mixed case.

Note: Once you enter SET CMD UPPER YES or SET CMD UPPER NO, you return to the Main Menu, no matter where you entered the command from. This is due to the fact that once the command is issued, CA Ideal “starts fresh” with the set of panels in the selected case.

SET COMPILE Command

This command sets various options for compilation listings.

This command has the following format:

```

      { ADV { YES }           }
      {   { NO }             }
      {                       }
      { BOD { YES }          }
      {   { NO }             }
      {                       }
      { DBCS { YES }         }
      {   { NO }             }
      {                       }
      { EXD { YES }          }
      {   { NO }             }
      {                       }
SET [SITE] COMPILE { IDE { YES }           }
                   {   { NO }             }
                   {                       }
                   { LINELIMIT { nnnnnn }  }
                   {                   { 999999 } }
                   {                       }
                   { LSQL { YES }          }
                   {   { NO }             }
                   {                       }
                   { MEL { YES }           }
                   {   { NO }             }
                   {                       }
                   { PANEL { FULL }        }
                   {   { SHORT }          }
                   {   { NO }             }
                   {                       }
                   { REF { FULL }          }
                   {   { SHORT }          }
                   {   { NO }             }
                   {                       }
```

Operand Definitions

Operand	Description
ADV	<p>Establishes whether advisory messages appear in the compilation listing. Specify:</p> <p>YES Advisory messages appear in the compilation listing.</p> <p>NO Advisory messages are suppressed from appearing in the compilation listing.</p>
BOD	<p>Establishes whether the body of the program appears in the compilation listing. Specify:</p> <p>YES The body of the program (working data, parameter data, and procedure definition) appears in the compilation listing.</p> <p>NO The working data, parameter data, and procedure definition is suppressed from appearing in the compilation listing.</p>
DBCS	<p>Enables validation of double-byte character set literals. Specify:</p> <p>YES The compiler validates double-byte character set literals.</p> <p>NO Validation of double-byte character set literals is suppressed.</p> <p>For more information about double-byte character set support, see the <i>Administration Guide</i>.</p>
EXD	<p>Establishes whether the external data appears in the compilation listing. Specify:</p> <p>YES External data (dataview definitions, report definitions, panel definitions, and copied SQLCAs) appear in the compilation listing.</p> <p>NO External data are suppressed from appearing in the compilation listing.</p>
IDE	<p>Establishes whether identification information appears in the compilation listing. Specify:</p> <p>YES Program identification, resources, and environment sections appear in the compilation listing.</p> <p>NO Identification information are suppressed from appearing in the compilation listing.</p>
LINELIMIT	<p>Prevents the administrator from any online compilation larger than the number of lines of source specified. Specify:</p> <p>nnnnnn A six-digit number that specifies the allowable lines of source that you can specify for the online compilation.</p> <p>999999 The default; this value in effect disables LINELIMIT.</p>

Operand	Description
LSQL	Establishes whether the SQL statements generated for FOR statements appear in the compilation listing. Specify: YES The generated SQL statements appear in the compilation listing. NO The generated SQL statements are suppressed from appearing in the compilation listing.
MEL	Establishes whether the lines of the procedure that are in error are highlighted. Specify: YES The error lines in the original procedure definition are highlighted. NO The error lines in the original procedure definition are not highlighted.
PANEL	Establishes whether the compilation listing includes a listing of panel components. Specify: FULL A complete listing of panel components are generated. SHORT Only the Identification, Facsimile, and Summary associated with each panel are generated. NO No panel listings are generated. Note: The current SET COMPILE EXD option supersedes this option. When EXD is YES, the PANEL option is recognized. Otherwise, the PANEL option is ignored.
REF	Establishes whether a cross-reference listing is produced with the compile. You can specify this option for online compiles for any batch compiles. No cross-reference listings are generated in CICS. Specify: FULL A complete cross reference listing is generated. SHORT Those symbol names that are defined but not referenced are suppressed. NO The cross reference listing is suppressed.

SET DATAVIEW VERSION Command (NonSQL Dataviews)

This command establishes a default version for modeled or unmodeled dataviews used in the CATALOG command when an optional version clause is not specified. The SET DATAVIEW VERSION command also establishes the default version used for a modeled or unmodeled dataview specified as a resource of a program without an explicit version when the program is compiled.

This command has the following format:

```
SET [SITE] DATAVIEW VERSION version
```

Operand Definitions

Operand	Description
<i>version</i>	Version used as the default for CATALOG commands and when a program is compiled. For more information about valid versions to specify, see Using Version Clauses in CA Ideal Commands in the chapter "Preliminary Concepts".

SET DBSQL Command (CA Datacom SQL Access)

This command establishes default options for CA Datacom/DB access plans. These options are defaults used for creating new program environment components. Changing these options has no effect on programs already created. You can override these options before compiling a program using the program's environment fill-in or before rebinding a compiled program using the ALTER PROGRAM ENVIRONMENT command.

This command has the following format:

```

{      AUTH auth-id      }
{      }
{      CBSIO nnnnnn     }
{      }
{          { DB }      }
{          { ISO }     }
{      DATE { USA }    }
{          { EUR }     }
{          { JIS }     }
{      }
{          { P }       }
{ DECPOINT { C }      }
{      }
{          { CS }     }
{ ISOLATION LEVEL { R } }
{          { U }     }
{      }
SET [SITE] DBSQL {      { ANSI86 } }
{ MODE { Datacom } }
{          { FIPS }   }
{      }
{ OPTMODE optmode     }
{      }
{          { NON }} }
{ OPTMSGGS { PREP { DET }} }
{          { EXEC { SUM }} }
{ PRIORITY nn          }
{          { A }       }
{ STRDELIM { Q }      ]
{      }
{          { DB }     }
{          { ISO }     }
{ TIME { USA }        }
{          { EUR }     }
{          { JIS }     }
{      }
{          { SEC }     }
{ WAIT nnn { MIN }    }

```

Operand Definitions

Operand	Description
AUTH	<p>Establishes the authorization ID for the access plan. Specify:</p> <p>auth-id A one- to eight-character value.</p> <p>Note: You can enter this command as AUTHID <i>auth-id</i>, ATZ <i>auth-id</i>, or as AUTH <i>auth-id</i>. The AUTHID abbreviation is valid only for this command and is not a generally accepted abbreviation. Also, the first character of the <i>auth-id</i> must be a special character (@, #, \$, and so on). It cannot be a numeric digit. CA Datacom enforces this rule.</p>
CBSIO	<p>Determines the I/O limit interrupt value for SQL statements that creates a set. Specify:</p> <p>nnnnnn A value from 0-524287.</p>
DATE	<p>Displays the format for SQL date type items. Specify:</p> <p>DB CA Datacom installed default.</p> <p>ISO International Standards Organization: yyyy-mm-dd</p> <p>USA U.S. standard: mm/dd/yyyy</p> <p>EUR European standard: dd.mm.yyyy</p> <p>JIS Japanese Industrial Standard: yyyy-mm-dd</p>
DECPOINT	<p>Specifies the character used as the decimal point when data displays. This has no effect on how the data is stored. Specify:</p> <p>P Period (.) is used as the decimal point and comma is used as the digit separator. This is the default.</p> <p>C Comma (,) is used as the decimal point and period (.) is used as the digit separator.</p>
ISOLATION-LEVEL	<p>Establishes the degree to which a unit of recovery is isolated from the updating operations of other units of recovery. Specify:</p> <p>CS Cursor stability. (CS) is required if you are doing updates, deletes, or inserts. You can abbreviate the keyword ISOLATION-LEVEL to ISO.</p> <p>R Repeatable read.</p> <p>U No locks are acquired</p>
MODE	<p>Establishes the mode in which CA Datacom processes the program. Specify:</p> <p>ANSI or ANSI86 (they are the same)</p> <p>Datacom</p> <p>FIPS</p>

Operand	Description
OPTMODE	Establishes the mode in which CA Datacom optimizes the plan by ordering joins. Specify one of the following orders: PREP Order joins during bind processing. MAN Order joins as specified in FROM clauses. EXEC Order joins at execution time.
OPTMSG	Specifies the type of optimization messages CA Datacom produces during bind processing (PREP) or during execution (EXEC). Specify: NON None (default) DET Detail SUM Summary
PRIORITY	Specifies the priority of the SQL requests. Specify: nn A value from 1 through 15, where the lowest priority is 1 and the highest priority is 15.
STRDELIM	Specifies the character that delimits string values in all SQL statements. Specify: A Apostrophes (') delimit string values. This is the default. Q Quotation marks (") delimit string values.
TIME	Displays the format for SQL time type items. Specify: DB CA Datacom/DB installed default ISO International Standards Organization: hh.mm.ss USA U.S. standard: hh:mm AM or PM EUR European standard: hh.mm.ss JIS Japanese Industrial Standard: hh:mm:ss
WAIT	Determines the exclusive control wait limit. Specify: nnn The number of minutes or seconds from 1 to 120. Then specify: SEC for seconds or MIN for minutes. For example, for a ten-second wait: WAIT 10 SEC.

SET EDIT Command

This command establishes the default values used while editing.

This command has the following format:

```

{ { BOUNDS } { num-1 [ MAX ] }}
{ { COLUMNS } { [ num-2 ] }}
{
}
{ CASE { UPPER } }
{ { MIXED } }
{
}
{ CONTEXT n }
{
}
{ HIGHLIGHT { OFF } }
{ { ERRORS } }
{
}
SET (SITE) EDIT { MARGIN { RIGHT } }
{ { LEFT } }
{
}
{ MULTIPLIER { RIGHT } }
{ { LEFT } }
{
}
{ TRUNCATION { Y/ON } }
{ { N/OFF } }

```

Operand Definitions

Operand	Description
BOUNDS COLUMNS	Establishes the default column boundaries used when searching for a string, changing a string, or shifting data. Specify: num-1 The left-hand column that begins the range. num-2 The right-hand column that ends the range. MAX (Default) The farthest right-hand column.
CASE	Establishes whether text entered in a program, report, or dataview is translated into uppercase. Text for a program, report, or dataview includes literals and comments. Specify: UPPER Text is translated to uppercase. MIXED Text is not translated; it is left as it was entered.

Operand	Description
CONTEXT	Establishes the default number of context lines (lines retained on the screen for reference) appearing above and below the area on the screen opened by an INPUT command. Specify: n A value from 1 to 5.
HIGHLIGHT	Establishes whether lines in the program that had errors during compilation are highlighted. Specify: OFF Highlighting is turned off. ERROR[S] Lines containing errors are highlighted.
MARGIN	Establishes the position of the sequence number and command portion when an EDIT or DISPLAY panel displays. Specify: LEFT Sequence numbers display on the left RIGHT Sequence numbers display on the right.
MULTIPLIER	Establishes where to place the number that acts as a multiplier or replication factor for a line command. Specify: LEFT Multipliers are placed on the left of the command. RIGHT Multipliers are placed on the right of the command.
TRUNCATION	Establishes whether truncation can occur as a result of a SHIFT or CHANGE command. Specify: OFF or N Prohibits any CHANGE command that would cause data to overflow the current column boundaries. That line is scrolled to the top of the edit window. Stops any shift that would move data on a line past the current column boundaries when the data on that line reaches the column boundary. ON or Y If data overflows the column boundaries, the data is truncated at the column boundaries.

SET ENVIRONMENT Command

This command sets certain environment variables.

This command has the following format:

```

{ ACCOUNT-ID xxxx }
{ }
{ COMPILE-ID xxxx }
{ }
{ CURRENCY value }
{ }
{ DATEFOR pattern }
{ }
{ DB2PLAN-EXIT { exit-name } }
{ { NONE } }
{ }
{ DECIMAL { PERIOD } }
{ { COMMA } }
{ }
SET [SITE] ENVIRONMENT { EXECERROR { QUIT } }
{ { QUITIDEAL } }
{ { CONTINUE } }
{ }
{ FINAL-ID { xxxx } }
{ { NONE } }
{ }
{ GLOBAL-POOL nnnnnnn }
{ }
{ LINES nn }
{ }
{ PRINT-ID xxxx }
{ }
{ SIGNON-EXIT { exit-name } }
{ { NONE } }
{ }
{ SQL { DB } }
{ { DB2 } }

```

Operand Definitions

Operand	Description
SITE	Use the SITE option with the following options to set the option for the entire site: DATEFOR, DB2PLAN-EXIT, GLOBAL-POOL, LINES, SIGNON-EXIT, and SQL.
ACCOUNT-ID	Assigns different development or production activities to separate user-specified CICS transactions. This allows CICS performance analysis packages or transaction accounting packages to isolate transactions for charge-back and resource consumption analysis. It also allows DB2 users to select application plans associated with transaction IDs from in CA Ideal. Specify: xxxx A one- to four-character CICS transaction ID. In accordance with IBM standards, it is recommended that transaction IDs not begin with C. You must also specify each ACCOUNT-ID specified in a SET ENVIRONMENT command as the TRANSID parameter in a CICS DFHPCT entry. The entry should look exactly like the one distributed for TRANSID=SCFD, except for the TRANSID and PRTY.
COMPILE-ID	Assigns COMPILE tasks to a separate user-specified CICS transaction. This allows CICS performance analysis packages or transaction accounting packages to isolate COMPILE transactions for charge-back and resource consumption analysis. Specify: xxxx A one- to four-character CICS transaction ID. You must also specify each COMPILE-ID specified in a SET ENVIRONMENT command as the TRANSID parameter in a CICS DFHPCT entry. For DB2 , you must specify each COMPILE-ID specified in a SET ENVIRONMENT command in a CICS DFHRCT entry patterned after the one for SAST.
CURRENCY	Displays an alternate currency symbol in panels and prints it in reports generated by an application during the current session. This command does not affect the currency symbol as it is specified in the edit pattern of a report definition or panel field definition or in a \$EDIT function in the procedure of a CA Ideal program. Specify: value Any symbol.

Operand	Description
DATEFOR	<p>Establishes the date format the compiler uses. See the \$DATE function in Chapter 5 of the <i>Programming Reference</i> Guide for a chart containing edit pattern rules. Specify:</p> <p>pattern Any logical combination of the date patterns. You can change date formats for a report at runtime with the SET REPORT DATEFOR command and not the SET ENVIRONMENT DATEFOR command.</p>
DB2PLAN-EXIT	<p>Enables or disables a CA Ideal plan name exit program for the site or for the current session. This exit program can select or modify the name of the pla CA Ideal provides to DB2. Specify:</p> <p>exit-name The name of the plan name exit program to execute.</p> <p>NONE Disables the plan name exit program.</p>
DECIMAL	<p>Uses an alternate convention for the digit separator and decimal point when displayed in panels or printed in reports generated by an application running during the current session. Specify:</p> <ul style="list-style-type: none"> ■ <code>,.</code> (Period) The decimal symbol is a period and the digit separator is a comma. ■ <code>,</code> (Comma) The decimal symbol is a comma and the digit separator is a period. <p>You cannot specify any other symbols.</p> <p>This command does not affect the decimal symbol as it is specified in the edit pattern of a report definition or panel field definition or in a \$EDIT function in the procedure of a CA Ideal program. For example, if X is defined as:</p> <pre>SET X = \$EDIT(N, PIC='ZZZ,ZZZ.99')</pre> <p>and the following command is entered</p> <pre>SET ENVIRONMENT DECIMAL ,</pre> <p>at runtime, X displays as 2.345,00.</p> <p>For dataviews containing fields that have been defined in Datadictionary with a European format edit pattern, it is necessary to execute the "SET ENVIRONMENT DECIMAL ," command before cataloging the dataview. The CA Ideal CATALOG dataview process will swap commas and periods to produce dataview objects with the CA Ideal default edit pattern.</p>

Operand	Description
EXECERROR	<p>Specifies whether to quit or continue after an error resulting from the execution of commands in a member. Including this command in a member affects all commands that follow it in the member until the next SET ENVIRONMENT EXECERROR command or until the end of the member. Specify:</p> <p>QUIT Execution of the member stops when any of the statements following causes an error. This is the default.</p> <p>QUITIDEAL The CA Ideal session terminates when any of the statements following causes an error.</p> <p>CONTINUE Execution of the member continues when any of the statements following causes an error.</p> <p>For example, executing the following CA Ideal member duplicates a series of programs to the next version and compiles them. If an error is encountered, a message displays and processing continues.</p> <pre>SET ENVIRONMENT EXECERROR CONTINUE DUPLICATE PROGRAM DEMO1 TO NEXT VERSION COMPILE DEMO1 VERSION LAST DUPLICATE PROGRAM JECLUP TO NEXT VERSION COMPILE JECLUP VERSION LAST ...</pre> <p>However, SET ENVIRONMENT EXECERROR does not execute subsequent commands if there was a run time error. A run time error stops execution of the member even if CONTINUE was specified.</p>
FINAL-ID	<p>Specifies the next CICS transaction to schedule when the OFF command is executed (explicitly or implicitly) in CA Ideal. If you do not specify this command, the CA Ideal signoff logo displays. Specify:</p> <p>xxxx The one- to four-character transaction ID to schedule when the current session ends. In accordance with IBM standards, it is recommended that transaction IDs not begin with C.</p> <p>NONE The screen is cleared and the signoff logo displays when the current session ends.</p> <p>For use with CA Dataquery, issue the following command to log off CA Ideal and log on to CA Dataquery, bypassing both the CA Ideal logoff screen and the CA Dataquery logon screen. The CA Dataquery Main Menu displays.</p> <pre>SET ENV FINAL-ID DQRY</pre>

Operand	Description
GLOBAL-POOL	Obsolete tuning method related to RUN-STATUS. This command applies to the current CICS partition and any subsequent CICS partitions and should always be set to 0.
LINES	Specifies the maximum number of lines per print page used by the compiler and PRINT commands. It applies to print requests to network and system printers when printing entities, session options, and indexes. It also applies to compile listings and displays in batch. Specify: nn The number of lines per page. The minimum number of lines per page is 20; the maximum is 99.
PRINT-ID	Assigns PRINT tasks to a separate user-specified CICS transaction. This allows CICS performance analysis packages or transaction accounting packages to isolate transactions for charge-back and resource analysis. Specify: xxxx A one- to four-character CICS transaction ID. You must also specify each PRINT-ID specified in a SET ENVIRONMENT command as the TRANSID parameter in a CICS DFHPCT entry. For DB2 , you must specify each PRINT-ID specified in a SET ENVIRONMENT command in a CICS DFHPCT entry patterned after the one for SAST.
SIGNON-EXIT	Establishes or changes a signon exit program for a site. The exit specified in this program goes into effect after the next signon. Specify: exit-name The name of the exit program to execute if an exit program is specified for a transparent signon.
NONE	Disables the site-specified signon program.
SQL	Establishes the default type of database to use for SQL access when a site can access multiple database management systems using SQL. The CATALOG command uses it as its default for SQL dataviews. It is used as the default primary database for the program environment fill-in. Specify: DB CA Datacom SQL access DB2 DB2 access

Other Plan Considerations

If the plan name exit is enabled, CA Ideal calls it before the first SQL statement in a logical unit of work. That is, it is called before the first SQL statement at the beginning of each CICS transaction and before the first SQL statement following each database Commit.

You can embed the first SQL statement SQL, SQL generated by a FOR construct for a DB2 dataview, or SQL in a non-ideal subprogram. A Commit can be a PDL TRANSMIT, CHECKPOINT, or BACKOUT statement, or an SQL COMMIT or ROLLBACK statement.

Online, CA Ideal applications can switch plan names only if the RCT entry for the current transaction ID specifies PLNEXIT=YES and PLNPGME=@IADRCTX, the RCT exit supplied by CA Ideal.

For more information on plan name exits, see the *Administration Guide*.

SET OUTPUT Command

This command establishes default values for output, including number of copies printed, destination, and retention.

This command has the following format:

```

      { COPIES value                }
      {                             }
      {           { NETWORK name } }
      { DESTINATION { SYSTEM name } }
      {           { LIBRARY     } }
      {           { MAIL mail-id } }
      {                             }
      { DISPOSITION value          }
      {                             }
      { MAX nnnnn                  }
      {                             }
SET OUTPUT { OPTIONS                }
      {                             }
      { POSTMSG { YES }             }
      {           { NO  }             }
      {                             }
      { PROCEDURE                   }
      {                             }
      { RETENTION nn                }
      {                             }
      { WIDTH nnn                   }
      {                             }
```

Operand Definitions

Operand	Description
COPIES	Establishes the default number of copies to print on a system or network printer if you omit the COPIES=parameter in a DESTINATION clause. This command is ignored in batch. Specify: value A value from 1 through 99.
DESTINATION	Establishes the default printout destination as either a [assign the value for email in your book] ID, network printer, a system printer, or as the output library, if you omit the DESTINATION clause on a command. Specify: NETWORK name Network printer name SYSTEM name System printer name LIBRARY Output library MAIL 'email-id' [assign the value for email in your book] destination Systems that share ADROUT with CICS need two DESTINATION definitions for each NETWORK printer—one with the CICS printer ID and one with the VTAM network node name.
DISPOSITION	Establishes the default output disposition. You can override this default on a COMPILE, RUN, PRODUCE, or PRINT command. Specify: value A value of HOLD, RELEASE, or KEEP.

The following table shows the effects of specifying each of the disposition options on print requests issued either online or in batch.

Print Request Issued - Online		Print Request Issued - Batch		
Destination		Destination		
Output Disposition	Output Library	Sys/Net Printer or MAIL	Output Library	System Printer
Release	Output is placed in output library for browsing at the terminal.	Output is placed in output library and batch job is printed on a system or network printer. No copy is retained in output library.	Output is placed in output library for browsing at the terminal.	Output is printed on the system printer in batch. No copy is placed in the output library.

HOLD	Output is placed in output library and held until released (using ALTER OUT and PRINT OUT).	Not applicable.
KEEP	Output is placed in output library and batch job is printed on a system or network printer. Copy is retained in output library.	Not applicable.

Operand	Description
MAX	Sets the maximum number of lines for any output. Specify: nnnn A value from 1 through 64,000.
OPTIONS	Displays a fill-in where you can set all output options for this session (SESSION) or for the site (SITE). You must specify either SESSION or SITE.
POSTMSG	Specifies whether the CA Ideal message indicating successful completion of a command is suppressed. Specify: YES Displays the message. NO Suppresses the message. This is advantageous in a production environment.
PROCEDURE	Specifies the name of the cataloged procedure executed to print an output at a SYSTEM printer. Specify any valid cataloged procedure name. If you change this parameter, be sure to rename or COPY the installed cataloged procedure PSSUTIL to the same name.
RETENTION	Sets the default number of days an output can reside in the output library before being deleted. Specify: nn A value from 1 through 99, but not greater than the site maximum.
WIDTH	Sets the default width for all outputs generated. For reports defined in RDF, CA Ideal overrides the default width with the defined report width. Specify: nnn A number from 1 to 240.

SET PANEL Command

This command establishes the default characters used as delimiters and symbols on the panel layout.

This command has the following format:

```

{typesym                }
{ALLOWCURRSIGN { YES } }
{                    { NO } }
{ALLOWDIGSEP { YES } }
{                    { NO } }
{ALLOWEOF { YES } }
{                    { NO } }
{ALLOWMINSIGN { YES } }
{                    { NO } }
{CASE { UPPER } }
{    { MIXED } }
{    { DEFAULT } }
{    { NEUTRAL } }
{    { BLUE } }
SET {SITE] PANEL {COLOR { RED } }
{    { PINK } }
{    { GREEN } }
{    { TURQUOISE } }
{    { YELLOW } }
{    { WHITE } }
{    {HORIZONTAL } }
{    {VERTICAL } }
{COPY {COLVIEW } }
{    {NOHEADING } }
{    {SIDEVIEW } }
{    {TOP } }
{    {BOTTOM } }
{    {YES } }
{DECIMAL { PERIOD } }
{    { COMMA } }
{ERRORFILL x }
{    { N } }
{ERRORHANDLING { * } }
{    { H } }
{    { B } }
{IFATTRIBUTE [ UA ] }
{    [ UN ] [ H ]}
{    [ PA ] [ L ]}
{    [ PS ] [ I ]}
{    { SPACE[S] } }
{    { LOWVALUES } }
{INFILL { ZERO } }

```

```
{          { UNDERSCORE } }
{LAYOUT { NULLFILL } }
{          { BLANKFILL } }
{          { UPPER } }
{LAYOUTCASE { MIXED } }
{NONDISPLAY { SPACE[S] } }
{          { x } }
{          { SPACE[S] } }
{OUTFILL { LOWVALUE[S] } }
{          { SPACE[S] } }
{          { x } }
{PF13 { YES } }
{          { NO } }
{PF781011 { YES } }
{          { NO } }
{          { Z } }
{REQUIRED { YES } }
{          { NO } }
{SCROLL { YES } }
{          { NO } }
{TFATTRIBUTE [ UA ] }
{          [ UN ] [ H ] }
{          [ PA ] [ L ] }
{          [ PS ] [ I ] }
{WIDEOPTION { YES } }
{          { NO } }
{WIDTH { YES } }
{          { NO } }
```


Operand Definitions

Operand	Description
<i>typesym</i>	<p>Specifies the name of the symbol used in the panel layout whose function is assigned to the designated character. Valid values for <i>typesym</i> are:</p> <p>STARTSYM Start of field ENDSYM End of field DELSYM Delete field NEWSYM Add field COPYSYM Copy field to new location MOVESYM Move field to new location DESTSYM Mark destination of copy or move operation REPSYM Repeating field</p> <p>For value, specify any character.</p> <p>You can define the symbols for a session on the Panel Session Option fill-in and for an individual panel on the Panel Parameter fill-in and on the Panel Layout fill-in.</p>
ALLOWCURRSIGN	<p>Determines whether a currency symbol is allowed on input for numeric values. Specify:</p> <p>YES Allows a currency symbol on input. NO Does not allow a currency symbol on input.</p>
ALLOWDIGSEP	<p>Determines digit separator editing for numeric values. Specify:</p> <p>YES Allows digit separators on input. Digit separators are removed before presentation to the application procedure. NO Does not allow digit separators on input.</p>
ALLOWEOF	<p>Determines whether the EOF key can clear a field. Specify:</p> <p>YES EOF key clears fields. NO EOF key does not clear fields.</p> <p>For more information about using the EOF key on panel fields, see the <i>Creating Panel Definitions Guide</i>.</p>

Operand	Description
ALLOWMINSIGN	Determines minus sign editing. Specify: YES Allows the minus indication on input but removes the minus indication before presentation to the application procedure as a negative number. The minus indication can be on the left, right, specified by parentheses, or specified as a credit as: -34, 34-, (34), CR. NO Does not allow the minus indication on input.
CASE	Establishes whether the text entered in a panel field by users at runtime is converted to uppercase. Specify: UPPER Converts text to uppercase. MIXED Text remains as typed.
COLOR	Establishes the default panel field colors, when applicable. Specify: DEFAULT GREEN NEUTRAL TURQUOISE BLUE YELLOW RED WHITE PINK Note: The values NEUTRAL and DEFAULT mean that no extended attribute is sent to the 3270 data stream, resulting in colors based on field attributes: GREEN Unprotected lowlight BLUE Protected lowlight WHITE Protected highlight RED Unprotected highlight If the value is set to one of the colors, then all fields, regardless of attributes, are the specified color.

Operand	Description
COPY	<p>Establishes the default destination and format when dataview fields are copied into a panel definition using the CREATE command or the COPY prompter (available during panel definition processing). Specify:</p> <p>HORIZONTAL (H) Places the fields one after another across the screen on the line. If a field in its entirety does not fit on the current line, the field is placed in the first position of the next line.</p> <p>VERTICAL (V) Places the fields in columnar fashion, down the screen.</p> <p>COLVIEW (C) Places the headings on a line above the fields. If a field or its heading do not fit on their respective line, they are both placed on the next set of lines. The field name is used as the heading, since, in panel definition, information is obtained about the dataview fields from the cataloged version and not from the database.</p> <p>NOHEADING (N) No headings are included in the layout. This is the default.</p> <p>SIDEVIEW (S) Places the headings to the left of each field. If the field and the heading do not fit on the line, they are both placed on the next line. The field name is used as the heading, since, in panel definition, information is obtained about the dataview fields from the cataloged version and not from the database.</p> <p>TOP (T) Copies any dataview fields copied into a panel definition to the top of the panel.</p> <p>BOTTOM (B) Copies any dataview fields copied into a panel definition to the bottom of the panel.</p> <p>YES (Y) Re-establishes the default of side labels and addition at BOTTOM.</p> <p>An error occurs if the length of a field or a field and its heading does not fit on one panel line. An error also occurs if a dataview field being copied duplicates the name of a field already present in the panel definition. You can set heading specifications for individual fields and dataviews on the copy panel prompter. The specifications on the prompter override the values in effect only for the designated dataview fields. The SET PANEL COPY command overrides the system defaults for a session.</p>

Operand	Description
DECIMAL	Establishes the default symbol used to represent a decimal point. Specify: PERIOD (P) The period (.) represents the decimal point and the digit separator is the comma (,). In addition to the values shown, you can also use a period (.) to specify PERIOD COMMA (C) The comma (,) represents the decimal point and the digit separator is the period (.). In addition to the values shown, you can use a comma (,) to specify COMMA.
ERRORFILL	Establishes the default character that marks a field that redisplay after an erroneous entry. Specify: x Any character.
ERRORHANDLING	Establishes how, by default, an erroneous field entry is highlighted. Specify: NONE (N) No highlighting. ERRORFILL (*) Returns the panel with the erroneous field filled with the error fill character. HIGHLIGHT (H) Returns the panel with the erroneous entry intensified. BOTH (B) Indicates the same as H if the field has an illegal value, but the same as * if a required field is missing (default).

Operand	Description
IFATTRIBUTE	<p data-bbox="797 323 1393 443">Establishes the default screen attributes for panel fields defined for input. The installed default is UAL. You can specify combinations of the following attributes:</p> <p data-bbox="797 457 1393 520">UA (A, U) A field that can accept any characters. It is unprotected. U, A, and UA are synonyms.</p> <p data-bbox="797 535 1393 625">UN (N) A field that can accept only 327x numeric characters (0-9,...) on a data entry terminal. It is unprotected. N and UN are synonyms.</p> <p data-bbox="797 640 1393 760">PA (P) A protected field. You cannot modify or delete a protected field. The cursor does not skip a P or PA field. The cursor does not skip to the next unprotected field.</p> <p data-bbox="797 774 1393 865">PS (S, PN) A protected field that the cursor skips over (cannot be accessed). The cursor skips to the next unprotected field. PN, PS, and S are synonyms.</p> <p data-bbox="797 879 1393 907">H A field displayed with high intensity characters.</p> <p data-bbox="797 921 1393 949">L A field that displays with regular (low) intensity.</p> <p data-bbox="797 963 1393 1026">I An invisible field (input or text) in which the characters do not display.</p> <p data-bbox="797 1041 1393 1104">C The field to contain the cursor when the panel displays.</p> <p data-bbox="797 1119 1393 1201">E A field that is treated as if it were entered on the current transaction. If a field has an attribute of E, it is assumed that the user entered the value for that field,</p>

Operand	Description
	<p>even if the value was entered on a previous transaction or was not entered at all (a default value). C is only effective for one field on any panel.</p> <p>A protected field is skipped if the previous field is defined as PS or if the previous field is defined as PA and has an end-of-field character (;). The hardware recognizes the end-of-field character as an indicator to skip to the next unprotected field. You must specify the start-field character to mark the beginning of a field. You can terminate a field with the end-of-field character or the start-field character.</p> <p>If you do not specify a protection attribute (PROTECTED, SKIPPED, ALPHA or NUMERIC), the default value (UNPROTECTED ALPHA) is used. If you do not specify a HIGHLIGHT attributed (LOWLIGHT, HIGHLIGHT, or INVISIBLE), the default value (LOWLIGHT) is used.</p> <p>Note: You can specify the attributes listed in any order. Therefore, any combination of these attributes is accepted.</p>
INFILL	<p>Establishes the default character that pads the unused portion of an alphanumeric field on input. (Always zero for numeric fields.) Specify:</p> <p>SPACE[S] Space.</p> <p>LOWVALUES The lowest value in the alphanumeric collating sequence.</p> <p>ZERO Zero (automatic value for numeric field).</p> <p>UNDERSCORE Underscore.</p> <p>x Represents itself; numeric digits are permitted.</p>
LAYOUT	<p>Sets the fill mode during panel layout editing. Specify:</p> <p>NULLFILL The panel layout displays with the data on each row followed by nulls.</p> <p>BLANKFILL The panel layout displays with the data on each row padded with blanks to the specified panel width.</p>

Operand	Description
LAYOUTCASE	<p>Establishes whether text in a panel layout is translated to uppercase or remains mixed uppercase and lowercase. Specify:</p> <p>UPPER Text in the panel layout is translated to upper case.</p> <p>MIXED Text in the panel layout is not translated, but remains in mixed case.</p>
NONDISPLAY	<p>Defines the character sent to the screen when the application sends a non-displayable character. Allowable values are:</p> <p>SPACE[S] Space.</p> <p>x Represents itself. Any character is permitted.</p>
OUTFILL	<p>Establishes the default output fill character used when an entire field contains nulls. Specify:</p> <p>S, SPACE[S] Results in a space (x'40').</p> <p>L, LOWVALUE[S] Results in a null (x'00').</p> <p>U, UNDERSCORE[S] Results in an underscore (x'60').</p> <p>X Any other character represents itself. Digits are excluded.</p>
PF13	<p>Establishes whether CA Ideal-defined key assignments for PF1/PF13 and PF3/PF15 apply at runtime. Specify:</p> <p>YES PF key assignments for PF1/PF13 and PF3/PF15 apply at runtime.</p> <p>NO PF key assignments for PF1/PF13 and PF3/PF15 do not apply at runtime.</p> <p>YES means PF2/PF14 then RETURNS from the HELP screen invoked by PF1/PF13.</p>
PF781011	<p>Establishes whether CA Ideal-defined key assignments for PF7/19, PF8/20, PF10/22, and PF11/23 apply. Specify:</p> <p>NO The application procedure assigns PF keys.</p> <p>YES CA Ideal-defined PF key assignments apply.</p> <p>Z CA Ideal-defined wide panel PF key assignments apply.</p>
REQUIRED	<p>Establishes whether, by default, you must supply field values. Specify:</p> <p>YES You must supply field values.</p> <p>NO You can omit field values.</p>

Operand	Description
SCROLL	Specifies whether to process the application each time the scroll command is issued. Specify: YES Process the application each time the scroll command is issued. NO Do not process the application each time the scroll command is issued.

Operand	Description
TFATTRIBUTE	Establishes the default screen attributes for panel

Operand	Description
	<p>fields defined with text. The installed default is PSL. Specify:</p> <p>UA (A, U) A field that can accept any characters. It is unprotected. U, A, and UA are synonyms.</p> <p>UN (N) A field that can accept only 327x numeric characters (0-9,...) on a data entry terminal. It is unprotected. N and UN are synonyms.</p> <p>PA (P) A protected field. You cannot modify or delete a protected field. The cursor does not skip a P or PA field. The cursor does not skip to the next unprotected field.</p> <p>PS (S, PN) A protected field that the cursor skips over (cannot be accessed). The cursor skips to the next unprotected field. PN, PS, and S are synonyms.</p> <p>H A field displayed with high intensity characters.</p> <p>L A field that displays with regular (low) intensity.</p> <p>I An invisible field (input or text) in which the characters do not display.</p> <p>C The field to contain the cursor when the panel displays.</p> <p>E A field that is treated as if it were entered on the current transaction. If a field has an attribute of E, it is assumed that the user entered the value for that field, even if the value was entered on a previous transaction or was not entered at all (a default value). C is only effective for one field on any panel.</p> <p>A protected field is skipped if the previous field is defined as PS or if the previous field is defined as PA and has an end-of-field character (;). The hardware recognizes the end-of-field character as an indicator to skip to the next unprotected field. You must specify the start-field character to mark the beginning of a field. The end-of-field character or the start-field character can terminate a field.</p> <p>If you do not specify a protection attribute (PROTECTED, SKIPPED, ALPHA or NUMERIC), the default value (UNPROTECTED ALPHA) is used. If you do not specify a HIGHLIGHT attribute (LOWLIGHT, HIGHLIGHT, or INVISIBLE), the default value (LOWLIGHT) is used.</p> <p>Note: You can specify the attributes listed in any order. Therefore, any combination of these attributes is accepted.</p>

Operand	Description
WIDEOPTION	Controls wide panel support for the session. Specify: ON, Y, YES Enables wide panel support for the session. OFF, N, NO Disables wide panel support for the session.
WIDTH	Sets the default width for panels created and displayed during the rest of the session. Specify: Uses the current screen width. nnn A value between 80 and 236.

SET PANEL SESSION OPTIONS Command

Displays the PDF Panel Session Option fill-in used to establish parameters for the session. A SET PANEL command is available for every option that can be defined on this panel. For more information, see Notes on Panel Session Options.

Panel Session Options Considerations

The session options are divided into three categories:

- Field Level Options
- Panel Parameter Options
- Edit Session Options

The values established on the Panel Session Option fill-in override the system default values. These values, for the most part, are used when a new panel is created.

You can override the values for an edit session, individual panel, or field on other PDF fill-ins or by command. For example, you can toggle the fill mode, nullfill, or blankfill during layout editing. You can also redefine the panel width for each panel.

Any values you specify on this fill-in take effect when you press Enter as if you entered the related command.

You can store the values established on the Panel Session Options fill-in by specifying YES to the last prompt "Save as profile". The profile member is named PDF#ON by default. Each time you modify the values on this fill-in, you can update the member.

To automatically execute this member at signon, include EXECUTE PDF#ON in your signon procedure.

For more information on each field of the Panel Session Option fill-in, see the *Creating Panel Definitions Guide*.

SET PLAN MAXSQL Command (DB2 Only)

This command specifies the maximum number of SQL statements allowed in a CA Ideal SQL module. This maximum controls the size of each SQL module that CA Ideal generates as a result of the GENERATE PLAN command.

This command has the following format:

```
SET [SITE] PLAN MAXSQL nnnn
```

Operand Definitions

Operand	Description
<i>nnnn</i>	Maximum number of SQL statements allowed (1 - 9999).

SET REPORT Command

This command establishes default report formatting options.

This command has the following format:

```

{   CONTFOOT { Y }       }
{           { N }       }
{   CONTHDAD { Y }       }
{           { N }       }
{   DATEFOR date-pattern }
{           { NO }      }
{           { BC }      }
{           { BL }      }
{   DATEPOS { BR }       }
{           { TC }       }
{           { TL }       }
{           { TR }       }
{   GAP nn             }

SET [SITE] REPORT
{   LINES nnn          }
{   NULLSYM x          }
{           { D }        }
{   PAGEFMT { H }        }
{           { NO }       }
{           { BC }       }
{           { BL }       }
{   PAGEPOS { BR }       }
{           { TC }       }
{           { TL }       }
{           { TR }       }
{   SPACING n          }
{   WIDTH nnn         }

```

Operand Definitions

Operand	Description
CONTFOOT	<p>Specifies whether control break footings are printed. Specify:</p> <p>Y Indicates that control break footings print at the end of each control group.</p> <p>N Suppresses the printing of control break footings.</p>

Operand	Description
CONTHEAD	Indicates whether control break headings print. Specify: Y Indicates that control break headings print at the beginning of each control group. N Suppresses the printing of control break headings.
DATEFOR	Determines the format of the date (for example, 'MMDDYY'). Note: Although date-editing replaces the strings MMDDYY with the appropriate values, any other literal entered are accepted as simply output without substitution.
DATEPOS	Indicates the position of the date: Specify: NO None BC Bottom center BL Bottom left BR Bottom right TC Top center TL Top left TR Top right
GAP	Indicates the number of blank characters left between defined columns of detail lines. Specify: <i>nn</i> A value from 1 through 66. You can also specify the value A (Automatic).
LINES	Indicates the total number of lines per printed page of a report. Specify: <i>nnn</i> A value from 1 through 250.
NULLSYM	Establishes a single special character to represent null values. <i>x</i> Any valid character.
PAGEFMT	Indicates the format of page numbers. Specify: D Digits only H With hyphens P With a one- to eight-character prefix
PAGEPOS	Indicates the position of page numbers. Specify: NO None BC Bottom center BL Bottom left BR Bottom right TC Top center TL Top left TR Top right

Operand	Description
SPACING	Indicates the number of blank lines left between printed detail lines. Specify: <i>n</i> A value of 1, 2, or 3.
WIDTH	Indicates the number of characters per line in a report. Specify: <i>nnn</i> A value from 40 through 230.

SET RUN Command

This command establishes certain conditions that are in effect while a CA Ideal application is running.

This command has the following format:

```
{ $RC { ZERO } }
{      { KEEP } }

{ CBSTRACE { Y/YES } }
{          { N/NO } }

{ CLEAR { QUITRUN } }
{       { RESHOW } }

{ ERROR-PNL { NONE } }
{           { name nnn sys-id } }

{ LOOPLIMIT nnnnnnnn }

{ PACKAGE-SET collection-id }

{ PLAN { name } }
{      { DEFAULT } }

{ QUITIDEAL { YES } }
{           { NO } }

{ QUITMSG { YES } }
{         { NO } }

{ SQL { MIXED } }
{     { STATIC } }

{ STRNO nnn }

{ UPDATE { YES } }
{        { NO } }

{ { URT } name }
{ { FILETABLE } }

{ XA { ON } }
{   { OFF } }
```


Operand Definitions

Operand	Description
SITE	Establishes the option for the entire site. The SITE option is valid with the following options: \$SRC, CLEAR, ERROR-PNL, LOOPLIMIT, QUITIDEAL, QUITMSG, SQL, STRNO, URT, and XA.
\$SRC	Defines how the value of the \$SRC return code is set for each run. Specify: ZERO The value of the return code is set to zero at the beginning of each run. KEEP Keeps the return code value set during the execution of a previous program or CA Ideal command. Each run starts with the most recent return code value.
CBSTRACE	Controls whether an CA Datacom CBS trace of CA Ideal applications is turned on. If the command is not issued, the default is NO. Specify: YES/Y CBS trace is turned on. NO/N CBS trace is not turned on. For information on the CBS trace, see the <i>Problem Determination Guide</i> .
CLEAR	Specifies the action taken if you press the Clear when a program transmitted a panel. This command is only in effect while a CA Ideal application is running. During CA Ideal activities other than RUN, the Clear key continues to return the Main Menu. Specify: QUITRUN The Clear key ends the application. If SET RUN QUITIDEAL YES is in effect, this also ends the CA Ideal session; otherwise, the user returns to the Main Menu. RESHOW The Clear key redisplay the current panel in the same state in which it was most recently transmitted.
ERROR-PNL	Redisplay the panel in the event of a fatal CA Ideal internal error during a run (and if you specified SET RUN QUITIDEAL YES). You must specify the panel name, version number, and system. NONE Resets the option to the default panel. name VERSION nnn SYSTEM sys-id The name, the version number, and the system ID of the system containing the panel to display if a fatal error terminates a run.

Operand	Description
LOOPLIMIT	<p>Establishes a maximum for the number of times a test-status program loops through a FOR or LOOP construct. A runtime error occurs if this limit is exceeded. You can override this command for the current session to a higher or lower value. Specify:</p> <p>nnnnnnnn The maximum number of iterations (0 through 999,999,999) allowed through each FOR or LOOP construct.</p>
PACKAGE-SET	<p>Identifies the collection that contains the package into which a subsequently executed program is bound. Specify:</p> <p>collection-id A valid collection id.</p>
PLAN	<p>(DB2 Only) Establishes a default application plan name for a CA Ideal session or run. This plan name is used unless a new plan name specified in a CA Ideal application or a CA Ideal plan name exit program supersedes it. You can include the command in your jobstream or sign-on member or issue it at the terminal before running an application. You can use it in batch and online. Specify:</p> <p>name The one- to eight-character name of an application plan. CA Ideal plan names can be a maximum of seven characters long. Non-Ideal subprograms can use a plan name of up to eight characters.</p> <p>DEFAULT Sets the default plan name to the value in the IDOPTSCB. The installed value is IDP110DV.</p>
QUITIDEAL	<p>Allows sign off to take place automatically at the completion of the application in progress. If the command is not issued, the default is NO. Specify:</p> <p>YES At the conclusion of the current application, the CA Ideal session is terminated and a signoff panel other than the normal CA Ideal signoff panel displays. You can replace this signoff panel with a site-specified signoff panel. See the <i>Administration Guide</i> for more information.</p> <p>Using YES in batch causes unpredictable return codes. Online, use with SET COMMAND LINE 0.</p> <p>NO The session is not terminated at the conclusion of an application.</p>
QUITMSG	<p>Determines whether the following message prints at the end of a run:</p> <p>IDADRUNP01I: Run Completed, RC=0</p> <p>Specify:</p> <p>YES Message prints.</p> <p>NO Message is suppressed.</p>

Operand	Description
SQL	<p>(DB2 Only) Lets you specify at run-time whether an application running against a DB2 database can execute dynamic SQL mode or must execute only static SQL mode. Specify:</p> <p>MIXED (Default) Indicates that dynamic SQL mode is allowed. (This option requires that dynamic SQL be allowed on the plan parameters fill-in.)</p> <p>STATIC Indicates that only static SQL mode is allowed.</p>
STRNO	<p>(VSAM Only) Lets you specify how many VSAM strings are required to access a VSAM data set through nested FOR constructs. If this command is not issued, the default value is 1. Specify:</p> <p>nnn Number of VSAM strings required. This value can be any integer number from 1 to 255. Set it to the maximum level of nesting in any FOR construct in the applications run while this command is in effect.</p> <p>This command is required only for batch VSE applications that contain nested FOR constructs for the same VSAM data set. This command is ignored for CICS.</p> <p>This command is not required for batch z/OS environments (OS batch) since z/OS allocate strings dynamically. However, if an application contains nested FOR constructs, setting the STRNO to a number greater than 1 can improve performance. Increasing the value of STRNO increases storage requirements, therefore, do not set STRNO any higher than is required for the applications being run.</p>
UPDATE	<p>Determines whether updates execute or are suppressed for all dataviews (CA Datacom native and SQL access, DB2, sequential, and VSAM). This command controls access through both the FOR construct and embedded SQL. Specify:</p> <p>YES Allows updating of the database during execution.</p> <p>NO Prevents updating of the database.</p> <p>The SET RUN UPDATE NO command only suppresses data access commands that request the addition, update, or deletion of a record. All other processes, including such processes as allocation of sequential files, are still executed.</p> <p>This command is intended for debugging and testing a program without actually updating the data. Since programs function as if the update was successful, including display of any program messages indicating that the update was successful, do not use this command as a method of enforcing security or to make an updateable program non-updateable.</p>

Operand	Description
URT FILETABLE	<p>For batch CA Ideal programs, use this command before a run to select an alternate user requirements table. Specify:</p> <p>name The name of an alternate user requirements user table.</p> <p>For online CA Ideal programs, this is not necessary because another technique automatically selects file tables (see the CA Datacom/DB documentation.)</p> <p>Online, set the ACCESS= parameter to RAN (random). If the ACCESS= parameter is set to RANSEQ, you can use a single URT for both online and batch processing, but this incurs additional overhead for online processing.</p>
XA	<p>For a z/OS or VSE/ESA capable system, this command tells CA Ideal to use XA, either for the site or for the current session. CA Ideal then uses 31-bit addressing and storage above the line while running programs. The default is OFF until SET RUN XA ON is issued. Specify:</p> <p>ON Use z/OS.</p> <p>OFF Do not use z/OS.</p>

SET SCROLL Command

This command establishes the default means of scrolling forward and scrolling backward.

This command has the following format:

```

                {CURSOR}
SET SCROLL {FRAME }
    
```

Operand Definitions

Operand	Description
CURSOR	Line containing the cursor is positioned at the top for scroll forward, or at the bottom for scroll backward.
FRAME	An entire region's contents are scrolled, except for overlapping "context" lines.

SET SITE Commands

SET SITE commands establish and change default values for an entire site. Any default value set with a SET SITE command becomes the site default and remains in effect until it is reset with another SET SITE command or until an individual user overrides it temporarily with a SET command. Each CA Ideal SET option is installed with a default that the CA Ideal Administrator can override using a SET SITE command. The result then becomes the site default. Some site options can be set by individual commands; others by changing a value on a fill-in.

For consistency, a member containing all the current site options can be kept and executed after a new install.

A complete description of each SET SITE command's syntax is given in this guide under the basic SET Command. For example, the SET [SITE] EDIT command is explained completely under the SET EDIT command. Following is a list of all SET SITE commands.

Edit Options

For complete instructions, see the SET EDIT command.

This command has the following format:

```
SET (SITE) EDIT { BOUNDS      }  
                { CASE        }  
                { CONTEXT     }  
                { HIGHLIGHT   } value  
                { MARGIN      }  
                { MULTIPLIER  }  
                { TRUNCATION  }
```

Panel Definition Facility Options

For complete instructions, see the SET PANEL command.

```

                                     { typesym      }
                                     { ALLOWCURRSIGN }
                                     { ALLOWDIGSEP   }
                                     { ALLOWMINSIGN   }
                                     { CASE           }
                                     { COPY          }
                                     { DECIMAL        }
                                     { ERRORFILL      }
                                     { ERRORHANDLING  }
SET [SITE] PANEL { IFATTRIBUTE } value
                                     { INFILL        }
                                     { LAYOUT         }
                                     { LAYOUTCASE     }
                                     { NONDISPLAY     }
                                     { OUTFILL        }
                                     { PF13           }
                                     { PF781011      }
                                     { REQUIRED        }
                                     { SCROLL         }
                                     { SESSION OPTIONS }
                                     { TFATTRIBUTE    }
                                     { WIDEOPTION     }
                                     { WIDTH         }
```

Report Definition Facility Options

For complete instructions, see the SET REPORT command.

```

                                     { CONTFOOT     }
                                     { CONTHEAD     }
                                     { DATEFOR       }
                                     { DATEPOS       }
                                     { GAP           }
SET [SITE] REPORT { LINES    } value
                                     { NULLSYM      }
                                     { PAGEFMT      }
                                     { PAGEPOS       }
                                     { SPACING       }
                                     { WIDTH         }
```

Plan Definition Options

CA Datacom SQL

```

                { AUTH          }
                { CBSIO        }
                { DATE         }
                { DECPOINT     }
                { ISOLATION-LEVEL }
SET [SITE] DBSQL { MODE         } value
                { OPTMODE      }
                { OPTMSG      }
                { PRIORITY     }
                { STRDELIM     }
                { TIME         }
                { WAIT         }

```

DB2

```
SET [SITE] PLAN MAXSQL value
```

Compile Options

```

                { ADV {YES}      }
                {   {NO }       }
                { BOD {YES}      }
                {   {NO }       }
                { DBCS {YES}     }
                {   {NO }       }
                { EXD {YES}      }
                {   {NO }       }
SET [SITE] COMPILE { IDE {YES}  }
                  {   {NO }    }
                  { LINELIMIT {nnnnnn} }
                  {           {999999} }
                  { LSQL {YES}   }
                  {   {NO }     }
                  { MEL {YES}    }
                  {   {NO }     }
                  { PANEL {FULL } }
                  {   {SHORT}   }
                  {   {NO }     }
                  { REF {FULL }  }
                  {   {SHORT}   }
                  {   {NO }     }

```

Environment Options

For more information on SET [SITE] ENVIRONMENT options, see the SET ENVIRONMENT command.

```
                {DATEFOR    }
                {DB2PLAN-EXIT}
SET [SITE] ENVIRONMENT {GLOBAL-POOL } value
                {LINES      }
                {SIGNON-EXIT }
                {SQL        }
```

RUN Options

For more information on SET [SITE] RUN commands, see the SET RUN command.

```
                { $RC      }
                { CLEAR    }
                { ERROR-PNL }
                { LOOPLIMIT }
                { PACKAGE-SET}
                { PLAN     }
SET [SITE] RUN { QUITIDEAL } value
                { QUITMSG  }
                { SQL      }
                { STRNO    }
                { URT      }
                { XA       }
```

Miscellaneous Options

You can find information on the following options under each respective command.

```
SET [SITE] ASYNCMSG value
SET [SITE] CATALOG DATAVIEW VALIDATION value
SET [SITE] CHECK DUPLICATE USER value
SET [SITE] VERSION value
SET [SITE] DATAVIEW VERSION value
```

Session Control Facility Options

You can set up the Session Control Facility options using a fill-in. A CA Ideal Administrator accesses this fill-in with the following command:

```
SET COMMAND SITE [OPTIONS]
```

For more information on fill-in, see the *Administration Guide*.

Site Options for Output

You can set output options for the Print Subsystem using a fill-in. Access this fill-in with the command

```
SET OUTPUT SITE OPTIONS
```

For more information on fill-in, see the *Administration Guide*.

SET VERSION Command

This command establishes the default version for entity types other than modeled and SQL dataviews for commands where an explicit VERSION clause is not supplied. Also see the SET DATAVIEW VERSION command.

This command has the following format:

```
SET [SITE] VERSION version
```

Operand Definitions

Operand	Description
<i>version</i>	The version to use as the default for entities in commands where an explicit version clause is omitted. For information on valid versions to specify, see Using Version Clauses in CA Ideal Commands in the “Preliminary Concepts” chapter.

SIGNON Command

When required, the SIGNON command should be the first in any batch jobstream. Follow this command with any sequence of CA Ideal commands. If an error is encountered while executing one of the commands, the transaction containing the error is ended and the next command is executed.

This command has the following format:

```

      {USER }
[SIGNON] {PERSON} userid PASSWORD password

```

Note: The SIGNON keyword itself is optional.

For more information, see the “Establishing Signon Processing” chapter in the *Administration Guide*.

SPLIT Command

You enter the SPLIT command in one of three ways:

- With the cursor position in the command area,
- With the cursor position in the display area,
- Or with line specification.

The SPLIT command without arguments divides the last or only region of the display area and always applies to the last region displayed. If there are already two regions, then the SPLIT applies to the second region. The cursor in the display area marks the point where splitting occurs. Specifying SPLIT with the cursor positioned in the command area divides the regions evenly. SPLIT with line numbers can divide the second region unevenly. Splitting the screen is also explained in the *Working in the Environment* Guide.

Note: PF key actions apply for the region where the cursor is located.

This command has the following format:

SPLIT [n [m]]

Operand Definitions

Operand	Description
SPLIT	<p><i>With the cursor positioned in the command area</i>, divides the regions evenly. The minimum region size is nine lines. If there is not enough room on the screen for another region, the SPLIT is not applied and an error message is issued.</p> <p>The current contents of the display area appear in the first region. All current option settings for the first region are retained for the second region. If the content for the second region is not specified with a command, the Main Menu appears in the second region.</p>
	<p><i>With the cursor positioned in the display area</i>, divides the regions at the cursor position. For example, the following command with the cursor positioned on the twentieth line of the screen, splits the screen into regions at line 20, assuming enough lines remain to accommodate a second region.</p>
n	Specifies the new size of the existing, last, or only region (the number of lines).

Operand	Description
<i>m</i>	Specifies the size of the new region (the number of lines). If you specify <i>m</i> , it stipulates a minimum number of lines that must be available to a second region for the split to take effect. When you do not specify <i>m</i> , the second region occupies the remaining available lines.

For example, the command, the following command formats 20 lines in the first region and the remainder in a second region.

```
SPLIT 20
```

The following command formats 20 lines in the first region and a minimum of 10 lines in the second region.

```
SPLIT 20 10
```

If the number of lines specified is greater than the number of lines available, the SPLIT command is ignored and a message is issued.

Example

When there is more than one region, you can prefix commands with a region number and a space. The commands apply by default to region 1 if no prefix is specified. You can only execute the RUN command in the first (or only) region. To display dataview EMPLOYEE in one region and to edit program definition SAMPGM in a second region, when only one region exists, enter:

```
SPLIT  
1 DISPLAY DATAVIEW EMPLOYEE  
2 EDIT PROGRAM SAMPGM
```

You can also enter commands on multiple lines or delimit them on a single line. For example:

```
SPLIT  
2 EDIT PROGRAM SAMPGM; 1 DISPLAY DATAVIEW EMPLOYEE
```

SUBMIT Command

The following SUBMIT command or equivalent SUBMIT prompter submits a data member containing a batch jobstream or a series of data members that contain portions of a jobstream. Access the SUBMIT prompter by selecting Option 3 on the Process Program Menu or issuing the SUBMIT command.

This command has the following format:

```

      { *
SUBMIT {name-1 [USER user-id]} [name-2 [USER user-id]] ...
      [name-8 [USER user-id]]

```

Operand Definitions

Operand	Description
*	Submits the current member.
<i>name-1</i> name-2...name-8	From one to eight member names. If you specify more than one member, the contents of the members are first concatenated in the order the members are specified, and then submitted.
<i>user-id</i>	The 1- to 15-character user name or the one- to three-character short user ID of the user who owns the member. This option is required only if the member is not yours.

Example

The following command concatenates portions of a jobstream contained in three different members, and submits them for execution.

```
SUBMIT JOBJCL STRTJCL IDTBTCH
```

The member JOBJCL contains the jobcard, the member STRTJCL contains the JCL to invoke CA Ideal, and IDTBTCH contains the CA Ideal commands to run in batch.

SYSTEM Command

A system in CA Ideal is a collection of application programs and the developers and users associated with them. The functions CA Ideal provides for the definition and maintenance of system definitions are illustrated in the System Maintenance Menu. To access this menu, select Option 2 on the Administration Maintenance Menu or enter the following CA Ideal command.

This command has the following format:

```
SYSTEM
```

TIME Command

The TIME command displays the current date and time.

This command has the following format:

```
TIME
```

Example

```
IDADTIME01I - Current date and time 08/26/04 08:39:14
```

The date format option on the SET COMMAND SESSION OPTIONS fill-in determines the format of the date. You can also change the date format through the SET COMMAND DATEFOR command.

USER Command

The functions CA Ideal provides for the definition and maintenance of user definitions are found in the User Maintenance Menu. To access this menu, select Option 1 on the Administration Maintenance Menu or enter the following CA Ideal command:

This command has the following format:

```
USER
```

WORK Command

The following command or equivalent PF key displays the working data definition fill-in for the current program definition.

This command has the following format:

WORK

The working data definition fill-in names and describes data local to the application.

For more information on working data definition the fill-ins, see the *Creating Programs Guide*.

Chapter 3: Editing in CA Ideal

This chapter describes the editing features available in each facility of CA Ideal.

Editing in CA Ideal

Issuing any CA Ideal EDIT command begins an editing session. The EDIT PROGRAM PROCEDURE or EDIT MEMBER command, for example, displays data in the region, with a scale line across the top of the region and a six-digit sequence number and command field on the right-hand side or left-hand side, (see SET EDIT MARGIN) as shown below.

```
=>
-----
IDEAL: PROCEDURE DEFINITION  PGM EMPL (001) TEST                SYS: DEM  FILL-IN
Command.....1.....2.....3.....4.....5.....6.....7..
===== T O P =====
000100 <<PROCESS_PANEL>> PROC
000200     TRANSMIT DISPANEL
000300     SELECT
000400     WHEN $ENTER-KEY
000500     FOR EMPLOYEE
000600     WHERE EMPLOYEE.NUMBER = DISPANEL.NUMBER
000700     MOVE DISPANEL TO EMPLOYEE BY NAME
000800     WHEN NONE
000900     SET DISPANEL.MSG = 'EMPLOYEE DELETED'
001000     SET ATTR HIGHLIGHT TEMP ON DISPANEL. NUMBER
```

During an edit session, you can edit a component of a CA Ideal entity by using:

- 3270 operations
- Line commands
- Primary editing commands
- PF keys

The components of CA Ideal that you can edit or display and a summary of the type of editing possible in a component are described in the following table.

Component	Editing Using 3270 Hardware	Using Primary Commands	Line Commands
Program Definition			
Identification	Yes	SCROLL FORWARD/BACKWARD, TOP/BOTTOM	None
Resources	Yes	All commands in EDIT mode; SCROLL FORWARD/BACKWARD, TOP/BOTTOM, FIND, NEXT, INCLUDE, EXCLUDE, FIRST, LAST, PREVIOUS, POSITION commands in DISPLAY mode.	All line commands in EDIT mode except the PDL Language. Construct Templates; only the * line command in DISPLAY mode.
Procedure	Yes		All line commands in EDIT mode; only the * line command in DISPLAY mode.
		.All commands in EDIT mode; SCROLL FORWARD/BACKWARD, ACKWARD, TOP/BOTTOM, FIND, NEXT, INCLUDE, EXCLUDE, FIRST, LAST, PREVIOUS, POSITION commands in DISPLAY mode.	

Component	Editing Using 3270 Hardware	Using Primary Commands	Line Commands
Working Data	Yes	.	All line commands in EDIT mode except the PDL Language Construct Templates; only the * line command in DISPLAY mode.
Parameter Data	Yes	.	.
Environment	Yes	SCROLL FORWARD/BACKWARD, TOP/BOTTOM, LEFT/RIGHT; plus CHECKPOINT and ROLLBACK commands in EDIT mode.	None
Panel Definition			
	Yes	SCROLL FORWARD/BACKWARD, TOP/BOTTOM, LEFT/RIGHT; plus CHECKPOINT and ROLLBACK commands in EDIT mode.	None
Parameters	Yes	.	
Layout	Yes	SCROLL FORWARD/BACKWARD, TOP/BOTTOM, LEFT/RIGHT, POSITION; plus CHECKPOINT, ROLLBACK, and INPUT commands in EDIT mode.	
Summary	Yes	SCROLL FORWARD/BACKWARD, TOP/BOTTOM, POSITION; plus CHECKPOINT and ROLLBACK in EDIT mode.	

Component	Editing Using 3270 Hardware	Using Primary Commands	Line Commands
Field name	Yes	SCROLL FORWARD/BACKWARD, TOP/BOTTOM, LEFT/RIGHT; plus CHECKPOINT and ROLLBACK commands in EDIT mode.	None
Clarify	No	SCROLL FORWARD/BACKWARD, TOP/BOTTOM.	.
Report Definition			
Identification	Yes	SCROLL FORWARD/BACKWARD, TOP/BOTTOM, LEFT/RIGHT; plus CHECKPOINT and ROLLBACK commands in EDIT mode.	None
Parameters	Yes	.	.
Detail	Yes	All commands in EDIT mode; SCROLL FORWARD/BACKWARD, TOP/BOTTOM, FIND, NEXT, INCLUDE, EXCLUDE, FIRST, LAST, PREVIOUS, POSITION commands in DISPLAY mode.	All line commands in EDIT mode except the PDL Language Construct templates; only the * line command in DISPLAY mode.
Column	Yes	All commands in EDIT mode; SCROLL FORWARD/BACKWARD, TOP/BOTTOM, FIND, NEXT, INCLUDE, EXCLUDE, FIRST, LAST, PREVIOUS, POSITION commands in DISPLAY mode.	All line commands in EDIT mode except the PDL Language Construct templates; only the * line command in DISPLAY mode.
Heading	Yes	All commands in EDIT mode; SCROLL FORWARD/BACKWARD, TOP/BOTTOM, FIND, NEXT, INCLUDE, EXCLUDE, FIRST, LAST, PREVIOUS, POSITION commands in DISPLAY mode.	All line commands in EDIT mode except the PDL Language Construct templates; only the * line command in DISPLAY mode.
Member	Yes	.	.
Display Dataview	No	SCROLL FORWARD/BACKWARD, TOP/BOTTOM, LEFT/RIGHT, FIND, NEXT, INCLUDE, EXCLUDE, FIRST, LAST, PREVIOUS, and POSITION commands only.	Only the * line command.

Component	Editing Using 3270 Hardware	Using Primary Commands	Line Commands
Display Output	No	SCROLL FORWARD/BACKWARD, TOP/BOTTOM, LEFT/RIGHT, FIND, NEXT, INCLUDE, EXCLUDE, FIRST, LAST, PREVIOUS, and POSITION commands only.	.
Display Index	No	.	None
Dataview Definition			
Identification	Yes	SCROLL FORWARD/BACKWARD, CHECKPOINT and ROLLBACK commands in EDIT mode.	None
Parameters	Yes	.	.
Fields	Yes	All commands in EDIT mode; SCROLL FORWARD/BACKWARD, TOP/BOTTOM, FIND, NEXT, INCLUDE, EXCLUDE, FIRST, LAST, PREVIOUS, POSITION commands in DISPLAY mode.	All Line commands in EDIT mode except the PDL Language Construct Templates; only the * line command in DISPLAY mode.
Keys	Yes	.	.
Package Definition			
Identification	Yes	SCROLL FORWARD/BACKWARD, TOP/BOTTOM, LEFT/RIGHT; plus CHECKPOINT and ROLLBACK commands in EDIT mode.	None
Parameters	Yes	.	.
Resource	Yes	All commands in EDIT mode; SCROLL FORWARD/BACKWARD, TOP/BOTTOM, FIND, NEXT, INCLUDE, EXCLUDE, FIRST, LAST, PREVIOUS, POSITION commands in DISPLAY mode.	All line commands in EDIT mode except the PDL Language Construct templates; only the * line command in DISPLAY mode.

Component	Editing Using 3270 Hardware	Using Primary Commands	Line Commands
Plan Definition			
Identification	Yes	SCROLL FORWARD/BACKWARD, TOP/BOTTOM, LEFT/RIGHT; plus CHECKPOINT and ROLLBACK commands in EDIT mode.	None
Parameters	Yes	.	.
Resource	Yes	All commands in EDIT mode; SCROLL FORWARD/BACKWARD, TOP/BOTTOM, FIND, NEXT, INCLUDE, EXCLUDE, FIRST, LAST, PREVIOUS, POSITION commands in DISPLAY mode.	All line commands in EDIT mode except the PDL Language Construct templates; only the * line command in DISPLAY mode
DBRM		.	.

Editing Using 3270 Hardware Facilities

Editing keys affect the field where the cursor is found. Each line in the region can contain various fields. For example, each line of a program working data definition contains seven fields. Each line of a program procedure contains two fields-the text field and the sequence number and command field.

ERASE EOF Key

When the cursor is in a display area field, erases data following the cursor to the end of the field.

Erasing a sequence number with the EOF key has no effect. If you erase the entire text field on a line with the EOF key, any line commands specified in the sequence number and command field on the line are ignored.

DELETE Key

Deletes characters and spaces one at a time at the cursor, while shifting data from the right and leaving null characters in the right-most columns of the field.

INSERT Key

Lets you enter characters in a field in the column containing the cursor. Each character entered shifts existing data one column to the right and replaces a null character to the right of the cursor until no null characters remain in that field.

Alphanumeric and special character keys

Enters the character represented on the key into the field in the column containing the cursor.

Tab keys

- **Forward Tab**-Positions the cursor to the beginning of the next field.
- **Back Tab**-Positions the cursor to the beginning of the current field or, if the cursor is at the beginning of the current field, the back tab positions the cursor to the beginning of the previous field.

Editing Using Primary Editing Commands

This section describes the CA Ideal primary editing commands in alphabetical order. You can enter one or more primary editing commands in the command area of the screen at one time. You can use it with line commands and PF keys.

Editing Using Line Commands

This section also describes the CA Ideal line commands that perform editing functions on individual lines or on a range of lines. Enter line commands in the six-character sequence number field on the right or left of the screen. They can overlay the sequence number anywhere in the sequence number field. Edit multipliers are numbers attached to line commands that specify how many times to perform the editing function.

The following rules apply to using line commands.

- Symbols entered in the sequence number field that are not valid line commands result in an error. For example, the following command results in an error because QQ is not a valid line command.

00Q100
- A line can only contain one line command; however, up to 15 lines in the data can contain line commands.
- You can use line commands to edit a single line or a range of lines (two or more lines). For single lines, enter the line command in the sequence number field on the line to be affected. For a range of lines, line commands mark the beginning and end of the range (included in the range are the lines containing the line commands).
- After you specify the start of a range of lines with a line command, you can perform scrolling and other line commands before you specify the end of the range. A PENDING message displays on the message line for the line command that is pending. This message remains until a range is fully specified or, in the case of COPY and MOVE, you specify the destination. You can cancel a line command that is pending by the RESET line command, deleting the command, or typing over the command. RESET also cancels all other pending line commands from the data.
- The following rules and examples describe how CA Ideal distinguishes between multipliers attached to line commands and digits remaining in the sequence number field.
 - Ignore leading zeros and leading blanks.
 - If the cursor is not in the sequence number field, ignore digits that were not changed (that is, digits that are in the same location and have the same value as before the line command was entered).

For example, the following line commands delete line number 003100:

```
D03100
00D100
003D00
00310D
```

The SET EDIT MULTIPLIER command determines where to place the multiplier for a line command, to the right or to the left. When SET EDIT MULTIPLIER equals left, the same rules hold. For example, the following line commands repeat line 003100 three times:

```
3R3100
03R100
```

However, 003R00 only repeats the line once; the 3 is ignored because it did not change the digit 3 from the sequence number.

When SET EDIT MULTIPLIER equals right, if the cursor is not in the sequence number field, the same rules hold. For example, the following line command entries are correct for repeating line 003100 four times:

```
R43100
0R4100
```

However, line commands are cursor sensitive when SET EDIT MULTIPLIER equals right. That is, CA Ideal interprets a line command to include all numbers between the line command and the cursor. In the following example, line 400 is replicated four times. (The cursor position is underlined.)

```
000100
000200
000300
00R400
```

In the next example, line 400 is replicated forty times.

```
000100
000200
000300
00R400
```

Remember, line commands with the multiplier on the left are not cursor sensitive.

CA Ideal works this way because of the way in which the 3270 terminal works. When CA Ideal receives the panel with your editing commands, the 3270 returns the contents of the sequence number fields and whether the fields changed. This introduces an ambiguity-if the digits in the sequence number area following your line command are not changed, is that because the digits were not typed or were the same digits typed over what was already there? If the sequence number field contains 000400, was the 40 typed or not?

CA Ideal has to make an assumption, and the assumption is that:

- If you typed a multiplier with a line command, your cursor is immediately to the right of the last digit you typed.
- If you did not type a multiplier, then your cursor is immediately to the right of the line command or not in the sequence number area.

This usually works fine. However, you might, for example, type a 4 and inadvertently move the cursor one space to the right, giving you 40 repetitions of your line instead of 4. To help protect against such mistakes, CA Ideal imposes a limit of 100 as a multiplier, and does not allow multipliers on the DELETE, COPY, and MOVE commands.

Order of Command Processing

Commands are processed in the following order:

1. F1/13 (Help) or F3/15 (Print Screen)
2. RESHOW
3. IGNORE
4. Updated text
5. RESET
6. Line commands, in top-down order
 - D
 - DD
 - DT
 - DB
 - M
 - MM
 - R
 - RR
 - C
 - CC
 - PDL language construct templates
 - A
 - B
 - *
 - I
 - IB
7. Primary commands
8. Other function keys

CHANGE Command

The CHANGE command changes occurrences of a character or a string of characters to another character or string of characters. You can use this command when editing certain components (see the last note at the end of this command description).

You can restrict CHANGE to an inclusive range of lines and columns with the LINES and COLUMNS operands or by using the SET EDIT BOUNDS command. You can also change a specific number of occurrences of a character or a string.

This command has the following format:

```
CHANGE [ALL] [FORWARD ] /string-a/string-b/  
      [n ] [BACKWARD]  
      [LINE[S] start-line [end-line]]  
      [COLUMNS start-column [MAX      ]]  
      [                [end-column]]
```

Operand Definitions

Operand	Description
ALL <i>n</i>	Number of occurrences of the string to change. ALL is the default.
FORWARD	Default. The search for string-a begins with the first line of the component or the specified start line and processes through subsequent lines.
BACKWARD	The search for string-a begins with the last line of the component or the specified end line and processes backward.
/	Character that delimits the strings in the command. You must use the same character consistently in a command. You can use any special character except the currently defined command delimiter (installed default is a semicolon (;)) or comment character (installed default is a colon (:)), an asterisk (*), an at sign (@), a pound sign (#), or dollar sign (\$).
<i>string-a</i>	Character or string of characters to change to string-b. <i>String-a</i> must be entirely in the column range and the string cannot include the string delimiter character or the current command delimiter.

Operand	Description
string-b	Character or string of characters that replaces the character or string of characters in string-a. You can specify an empty <i>string-b</i> by placing two delimiters right next to each other, with no intervening spaces, for example, CHANGE <i>/string-a//</i> . This deletes <i>string-a</i> . The string cannot include the string delimiter character or the current command delimiter.
start-line	Line where the search for <i>string-a</i> begins. You can specify it as: num The sequence number of the start line. TOP Indicates the first line of the entity. If a start line is not specified, TOP is the default. CURSOR Indicates the position where the cursor is located. offset A position relative to the top line displayed: * The top line displayed. *+n The line n number of lines below the top line displayed. *- n The line n number of lines before the top line displayed. Note: You can omit the keyword LINE when you specify TOP, BOTTOM, or an offset.
end-line	Line where the search for string-a ends. You can specify it as: BOTTOM Default. Last line of the entity. num Sequence number. CURSOR Indicates the position where the cursor is located.
offset	Position relative to the top line displayed: * Top line displayed. *+ n Line n number of lines below the top line displayed. *-n The line n number of lines before the top line displayed.
start-column	Column where the search for string-a begins.
MAX	Default. The farthest right-hand column.
end-column	Column where the search for string-a ends.

- Because TOP and BOTTOM are the default limits on a range of lines, specifying the following changes the top of the entity through the bottom.

CHANGE */string-a/string-b/*

- Specifying the following changes lines 200 through the bottom of the entity.

CHANGE */string-a/string-b/* LINE 200

- To change a single line by sequence number, specify the sequence number as the start line and end line. For example:

```
CHANGE /string-a/string-b/ LINE 200 200
```

- Specifying the following only changes the top line:

```
CHANGE /string-a/string-b/ * *
```

- You can specify line and column clauses in either order. For example:

```
CHANGE /DISPANEL/DISPNL/ LINES 165 175 COLUMNS 10 40
```

or

```
CHANGE /DISPANEL/DISPNL/ COLUMNS 10 40 LINES 165 175
```

- The column range can be the entire line, a range specified with the command SET EDIT BOUNDS, or an explicit range specified with CHANGE.
- If string-b is longer than string-a, then string-b replaces string-a including trailing blanks up to the length of string-b. If there are not enough trailing blanks in the column range, the current setting of SET EDIT TRUNCATION controls the result.
 - If SET EDIT TRUNCATION is ON, data is truncated from the left starting at the end of the column range.
 - If SET EDIT TRUNCATION is OFF, the change is not made.

For instance, in the following example, ABCbDEF and BCbbDEF are in columns 1-7 (b represents a blank). The command, CHANGE /C/123/ COL 1 4 results in the following:

Before	After (Truncation On)	After (Truncation Off)
ABCbDEF	ABCbDEF	AB123EF
BCbbDEF	B123DEF	B123DEF

- If the change cannot be made, a message displays and the affected line is brought to the top of the region. All changes before that line are kept, but no changes are made to the affected line.
- If SET EDIT BOUNDS is in effect and you use the CHANGE command without specifying column boundaries, the following defaults are used.

	Column Edited	Column Boundaries
Member	Data area	1 - 72
Report		
Detail	Field name	1-25
Heading	Field name	1-56

	Column Edited	Column Boundaries
Column	Field name	1-25
Program		
Procedure	Data area	1-72
Parameter	Field name	1-19
Working Data	Field name	1-19
Dataview		
Field	Field name	1-19
Key (VSAM)	Field name	1-32
Plan		
DBRM	DBRM	1-8
Resource	Program name	1-8

Example

The following example illustrates the use of the CHANGE command to change all occurrences of W#EMPLOYEE to W#EMPDATA. The COLUMNS operand restricts the change to column 40 through the farthest right-hand column.

```

=> CHANGE /EMPLOYEE/EMPDATA/ COLUMNS 40

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION  PGM EMPLST (001) TEST          SYS: DEM  FILL-IN

Command....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
000001 FOR EACH EMPLOYEE
000002   WHERE where_condition
000003   MOVE EMPLOYEE.FLD1          TO   W#EMPLOYEE.FLD1
000004   MOVE EMPLOYEE.FLD2          TO   W#EMPLOYEE.FLD2
000005   MOVE EMPLOYEE.FLD3          TO   W#EMPLOYEE.FLD3
000006   MOVE EMPLOYEE.FLD4          TO   W#EMPLOYEE.FLD4
000007   MOVE EMPLOYEE.FLD5          TO   W#EMPLOYEE.FLD5
000008   MOVE EMPLOYEE.FLD6          TO   W#EMPLOYEE.FLD6
000009   MOVE EMPLOYEE.FLD7          TO   W#EMPLOYEE.FLD7
000010   MOVE EMPLOYEE.FLD8          TO   W#EMPLOYEE.FLD8
000011   MOVE EMPLOYEE.FLD9          TO   W#EMPLOYEE.FLD9
000012   MOVE EMPLOYEE.FLD10         TO   W#EMPLOYEE.FLD10
000013 WHEN NONE
000014   statements
000015 ENDFOR

```

The results of the CHANGE command follow.

```

=> Changed 10 field(s) in 10 record(s)

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION  PGM EMPLST (001) TEST          SYS: DEM  FILL-IN

Command....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
000001 FOR EACH EMPLOYEE
000002     WHERE where_condition
000003     MOVE EMPLOYEE.FLD1          TO      W##EMPDATA.FLD1
000004     MOVE EMPLOYEE.FLD2          TO      W##EMPDATA.FLD2
000005     MOVE EMPLOYEE.FLD3          TO      W##EMPDATA.FLD3
000006     MOVE EMPLOYEE.FLD4          TO      W##EMPDATA.FLD4
000007     MOVE EMPLOYEE.FLD5          TO      W##EMPDATA.FLD5
000008     MOVE EMPLOYEE.FLD6          TO      W##EMPDATA.FLD6
000009     MOVE EMPLOYEE.FLD7          TO      W##EMPDATA.FLD7
000010     MOVE EMPLOYEE.FLD8          TO      W##EMPDATA.FLD8
000011     MOVE EMPLOYEE.FLD9          TO      W##EMPDATA.FLD9
000012     MOVE EMPLOYEE.FLD10         TO      W##EMPDATA.FLD10
000013     WHEN NONE
000014     statements
000015 ENDFOR

```

The following example illustrates the use of the FIND and CHANGE commands to find all occurrences of ASK and change all occurrences of RLSASK to RLSDIS.

```

=> FIND ALL /ASK/

-----
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: DEM  FILL-IN

Command....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
000001 <<ASK>> PROCEDURE
000002     IF EXIST
000003         MOVE $SPACES TO RLSASK.MSG
000004         LOOP
000005         TRANSMIT RLSASK CLEAR
000006     WHILE $PANEL-ERROR
000007         ENDL00P
000008     ELSE
000009         TRANSMIT RLSASK
000010         IF RLSASK.NUMBER = 99999
000011             QUIT RUN
000012         ENDIF
000013         SET EXIST EQ TRUE
000014     ENDIF
000015     DO DIS-ASK

```

Using the display of lines containing ASK, the CHANGE is applied to the lines specified.

```
=> CHANGE /ASK/DIS/ LINES 3 10
```

```
-----
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: DEM  FILL-IN
INCL /ASK/
Command.....1.....2.....3.....4.....5.....6.....7..
000001 <<ASK>> PROCEDURE
000003     MOVE $SPACES TO RLSASK.MSG
000005     TRANSMIT RLSASK CLEAR
000009     TRANSMIT RLSASK
000010     IF RLSASK.NUMBER = 99999
000015     DO DIS-ASK
```

The results of the CHANGE follows:

```
1 - Changed 4 field(s) in 4 record(s)
-----
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: DEM  FILL-IN
Command.....1.....2.....3.....4.....5.....6.....7..
===== T O P =====
000001 <<DIS>> PROCEDURE
000002     IF EXIST
000003     MOVE $SPACES TO RLSDIS.MSG
000004     LOOP
000005     TRANSMIT RLSDIS CLEAR
000006     WHILE $PANEL-ERROR
000007     ENDL0OP
000008     ELSE
000009     TRANSMIT RLSDIS
000010     IF RLSDIS.NUMBER = 99999
000011     QUIT RUN
000012     ENDIF
000013     SET EXIST EQ TRUE
000014     ENDIF
000015     DO DIS-ASK
```

CHECKPOINT Command

The CHECKPOINT command is used in an editing session to establish a new stable point where the user is returned if a ROLLBACK command is issued.

At the start of every editing session, CA Ideal makes a temporary copy of the component being edited. This copy is used as the initial automatic checkpoint. A CHECKPOINT issued during an editing session replaces the existing temporary copy of the entity with a copy that reflects any changes made to the entity during the editing session up to the time the CHECKPOINT command is issued. This new copy is then the checkpoint. Only the CHECKPOINT and ROLLBACK commands use the temporary copy. Editing activities affect the actual entity after you press the Enter key. At the conclusion of the editing session, CA Ideal deletes the temporary checkpoint copy.

When editing the working data, parameter data, and procedure of a CA Ideal program definition, a CHECKPOINT applies only to that component. For report and panel definitions, the entire definition is included in a CHECKPOINT, regardless of the component being edited when the CHECKPOINT is issued.

CHECKPOINT merely establishes a backup copy. To access that copy after editing activity, you must use ROLLBACK. You can use the DISPLAY ROLLBACK command at any time during an editing session to view the temporary checkpoint copy.

This command has the following format:

```
CHECKPOINT
```

COPY Command

The COPY command copies all or part of the working data, parameter data, or procedure of a CA Ideal program definition, or a member.

If you specify neither a program nor member name, data is copied from one specified destination in the current program or member to another specified destination in the current program or member.

This command has the following format:

```
[ [ {Pnnn} ] ]
[PROGRAM name [VER {PROD}] [SYS id]]
[ [ {LAST} ] ]
COPY [ ] [num-1 [num-2 ]]
[ ] [TOP [BOTTOM]]
[MEMBER name [USER {name}] ]
[ [ {id } ] ]
{num-3 }
{TOP }
{BOTTOM}
```


Operand Definitions

Operand	Description
PROGRAM <i>name</i>	One- to eight-character name of the program definition from which data is copied. The component of the program definition from which data is copied corresponds to the component being edited in the current program. For example, if you issue the COPY command while the procedure of the current program is being edited, the data is copied from the procedure of the program definition named in the COPY command. This applies to the working data and parameter data components too.
VERSION	Version of the program definition from which data is copied. nnn One- to eight-character number CA Ideal assigned when this version of the program was created. PROD Production-status version. LAST Latest version of the program created; for example, the version with the highest version number.
SYS <i>id</i>	System where the program resides. This is required when copying from a different system.
MEMBER <i>name</i>	One- to eight-character name of the member from which data is copied.
USER <i>name</i> USER <i>id</i>	The 1- to 15-character user name or the one- to three-character ID of the user who owns the member. This clause is required if the member does not belong to the current user.
<i>num-1</i> TOP	Line or the start of the range of lines to copy. num-1 Sequence number of the line to copy or, if specified with <i>num-2</i> or BOTTOM, the first line of a range of lines to copy. TOP Copies the first line in the program definition or member. If specified with <i>num-2</i> or BOTTOM, it indicates that the first line of a range of lines to copy is the top line. Note: If only one value is specified, it is the destination of the lines to copy. If two values are specified, they are the line to copy and the destination line.

Operand	Description
<i>num-2</i> <i>BOTTOM</i>	End of the range of lines to copy. num-2 Sequence number of the line marking the end of the range to copy. BOTTOM Indicates that the last line in the program or member is the end of the range of lines to copy. Note: If only one value is specified, it is the destination of the lines to copy. If two values are specified, they are the line to copy and the destination line.
<i>num-3</i> <i>TOP</i>	BOTTOM Destination of the lines to copy. The destination is always in the component being edited in the current program or the current member. num-3 Sequence number of the line in the component of the current program definition or the current member after which the line or lines being copied are inserted. TOP Indicates that the destination of the line or lines to copy is the top of the component of the current program definition or the current member. BOTTOM Indicates that the destination of the line or lines to copy is the bottom of the current program definition or member. Note: If only one value is specified, it is the destination of the lines to copy. If two values are specified, they are the line to copy and the destination line.

Copy Line Command

The copy line commands insert the designated line or range of lines at a location marked as the destination of the copy function and also retain the line or lines at the original location.

This command has the following format:

```
{C }  
{CC}  
  {A}  
[n] {B} [n]
```

Operand Definitions

Operand	Description
C	Specifies a single line to copy. You must specify a destination indicated by A or B with each C line command. The destination can occur before or after the C line command.
CC	<p>Used in pairs to indicate both the start and end of the range of lines to copy. You must specify a destination indicated by the line command A or B with each delimited range of lines. The destination can occur before or after the range.</p> <p>After specifying the start of a range with the CC line command, you can scroll the region forward or backward and perform other line commands.</p> <p>A (After) Copies a line or a range of lines immediately after the line containing the A line command.</p> <p>B (Before) Copies a line or a range of lines immediately before the line containing the B line command.</p> <p>n (Optional) Specifies the number of times (from 1 through 100) the line or range of lines is inserted at the specified location.</p>

The SET EDIT MULTIPLIER command determines whether *n* is placed to the left or right. For example:

If multiplier is set with:	then you specify:
SET EDIT MULTIPLIER RIGHT	A7 or B7
SET EDIT MULTIPLIER LEFT	7A or 7B

You can mark only one destination (for a move or a copy) at any one time.

The inserted lines are assigned sequence numbers in increments of one, starting from the previous existing line. When the number of inserted lines exceeds the number of integers between the sequence numbers of the existing lines, the sequence numbers on the lines following the insertion are changed.

Example

The following example illustrates copying a range of lines, 200 to 900, to a new location before line 1500.

```
=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION PGM EMPLST (001) TEST          SYS: DEM FILL-IN

Command.....1.....2.....3.....4.....5.....6.....7..
===== T O P =====
000100 SELECT
CC0200 WHEN $ENTER-KEY
000300 FOR EMPLOYEE
000400 WHERE EMPLOYEE.NUMBER = DISPANEL.NUMBER
000500 MOVE DISPANEL TO EMPLOYEE BY NAME
000600 WHEN NONE
000700 SET DISPANEL.MSG = 'EMPLOYEE DELETED'
000800 SET ATTR HIGHLIGHT TEMP ON DISPANEL.NUMBER
CC0900 ENDFOR
001000 SET NEXT_PANEL = 'DISPANEL'
001100 WHEN $PF11
001200 SET NEXT_PANEL = 'EMPMENU'
001300 WHEN $PF12
001400 DO TERMINATE_SES
B01500 WHEN OTHER
001600 SET DISPANEL.MSG = 'UNABLE TO INTERPRET KEY ENTERED'
001700 SET NEXT_PANEL = 'DISPANEL'
001800 ENDSEL
```

The original lines retain their position while a copy of the lines is inserted before line 1500.

```

=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION PGM EMPLST (001) TEST          SYS: DEM FILL-IN

Command.....1.....2.....3.....4.....5.....6.....7..
===== T O P =====
000100 SELECT
000200 WHEN $ENTER-KEY
000300 FOR EMPLOYEE
000400 WHERE EMPLOYEE.NUMBER = DISPANEL.NUMBER
000500 MOVE DISPANEL TO EMPLOYEE BY NAME
000600 WHEN NONE
000700 SET DISPANEL.MSG = 'EMPLOYEE DELETED'
000800 SET ATTR HIGHLIGHT TEMP ON DISPANEL.NUMBER
000900 ENDFOR
001000 SET NEXT_PANEL = 'DISPANEL'
001100 WHEN $PF11
001200 SET NEXT_PANEL = 'EMPMENU'
001300 WHEN $PF12
001400 DO TERMINATE_SES
001401 WHEN $ENTER-KEY
001402 FOR EMPLOYEE
001403 WHERE EMPLOYEE.NUMBER = DISPANEL.NUMBER
001404 MOVE DISPANEL TO EMPLOYEE BY NAME
001405 WHEN NONE
001406 SET DISPANEL.MSG = 'EMPLOYEE DELETED'
001407 SET ATTR HIGHLIGHT TEMP ON DISPANEL.NUMBER
001408 ENDFOR
001500 WHEN OTHER
001600 SET DISPANEL.MSG = 'UNABLE TO INTERPRET KEY ENTERED'
001700 SET NEXT_PANEL = 'DISPANEL'
001800 ENDSEL

```

DELETE Command

The DELETE command deletes a single line, a range of lines, or all of a member.

You must specify a beginning line, an ending line, or both. If you specify only the beginning or ending line of a range, only one line is deleted. If you specify only the ending line of a range, TOP is the default beginning line.

This command has the following format:

```

                [nnn ] [nnn ]
DELETE [LINES] [TOP ] [BOT ]

```

Operand Definitions

Operand	Description
<i>nnn</i>	Sequence number of the line where the delete begins or ends.
TOP	First line of the member.
BOT	Last line of the member. If only the start line of a range is specified, BOT is the default ending line.

Delete Line Command

The delete line command deletes a single line or a range of lines.

This command has the following format:

```
{ D}  
{DT}  
{DB}  
{DD}
```

Operand Definitions

Operand	Description
D	Entered on a line of data to delete. You can specify several D line commands on the same screen.
DT	Indicates the end of a range of lines to delete, starting with the first line of the member.
DB	Indicates the start of a range of lines to delete, ending with the last line of the member.
DD	Used in pairs to indicate the start and end of a range of lines to delete. The second occurrence of this command on the same or subsequent screen is interpreted as the end of the range. After specifying the start of a range with the DD line command, you can scroll the region forward or backward and can perform other line commands. Until the end of the range started by a DD line command is found, a PENDING message appears in the message line. When interpreting line commands that mark ranges of lines to delete, the commands are matched into pairs from the top down, regardless of the order in which you entered the line commands.

The sequence numbers of deleted lines are also deleted (see the following example). To renumber the sequence numbers and eliminate gaps in numbering caused by deletions, see the RENUMBER primary editing command.

Example

The following example illustrates deleting a single line, 000700, and a range of lines, 001200 to 001700.

```
=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION PGM EMPLST (001) TEST          SYS: DEM FILL-IN

Command.....1.....2.....3.....4.....5.....6.....7..
===== T O P =====
000100 SELECT
000200 WHEN $ENTER-KEY
000300 FOR EMPLOYEE
000400 WHERE EMPLOYEE.NUMBER = DISPANEL.NUMBER
000500 MOVE DISPANEL TO EMPLOYEE BY NAME
000600 WHEN NONE
D00700 SET DISPANEL.MSG = 'EMPLOYEE DELETED'
000800 SET ATTR HIGHLIGHT TEMP ON DISPANEL.NUMBER
000900 ENDFOR
001000 SET NEXT_PANEL = 'DISPANEL'
001100 WHEN $PF11
DD1200 SET NEXT_PANEL = 'EMPMENU'
001300 WHEN $PF12
001400 DO TERMINATE_SES
001500 WHEN OTHER
001600 SET DISPANEL.MSG = 'UNABLE TO INTERPRET KEY ENTERED'
DD1700 SET NEXT_PANEL = 'DISPANEL'
```

Once the data is applied, the designated lines are deleted. The sequence numbers on deleted lines are also deleted.

```
=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION PGM EMPLST (001) TEST          SYS: DEM FILL-IN

Command.....1.....2.....3.....4.....5.....6.....7..
===== T O P =====
000100 SELECT
000200 WHEN $ENTER-KEY
000300 FOR EMPLOYEE
000400 WHERE EMPLOYEE.NUMBER = DISPANEL.NUMBER
000500 MOVE DISPANEL TO EMPLOYEE BY NAME
000600 WHEN NONE
000800 SET ATTR HIGHLIGHT TEMP ON DISPANEL.NUMBER
000900 ENDFOR
001000 SET NEXT_PANEL = 'DISPANEL'
001100 WHEN $PF11
```

The following illustrates deleting lines 000800 to 001400, using the DB command.

```
=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION  PGM EMPLST (001) TEST          SYS: DEM  FILL-IN

Command.....1.....2.....3.....4.....5.....6.....7..
===== T O P =====
000100 SELECT
000200 WHEN $ENTER-KEY
000300 FOR EMPLOYEE
000400 WHERE EMPLOYEE.NUMBER = DISPLANEL.NUMBER
000500 MOVE DISPANEL TO EMPLOYEE BY NAME
000600 WHEN NONE
000700 SET ATTR HIGHLIGHT TEMP ON DISPANEL.NUMBER
DB0800 ENDFOR
000900 SET NEXT_PANEL = 'DISPANEL'
001000 WHEN $PF11
001100 WHEN $PF12
001200 DO TERMINATE_SES
001300 WHEN OTHER
001400 SET DISPANEL.MSG = 'UNABLE TO INTERPRET KEY ENTERED'
```

Once the DB command is applied, the designated lines are deleted as shown below.

```
=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION  PGM EMPLST (001) TEST          SYS: DEM  FILL-IN

Command.....1.....2.....3.....4.....5.....6.....7..
===== T O P =====
000100 SELECT
000200 WHEN $ENTER-KEY
000300 FOR EMPLOYEE
000400 WHERE EMPLOYEE.NUMBER = DISPLANEL.NUMBER
000500 MOVE DISPANEL TO EMPLOYEE BY NAME
000600 WHEN NONE
000700 SET ATTR HIGHLIGHT TEMP ON DISPANEL.NUMBER
```

Operand Definitions

Operand	Description
<i>nnn</i>	Sequence number of the line where the delete begins or ends.
TOP	First line of the member.
BOT	Last line of the member. If only the start line of a range is specified, BOT is the default ending line.

DISPLAY ROLLBACK Command

You can issue the DISPLAY ROLLBACK command at any time during an edit session to display the backup copy CA Ideal maintains for the current entity only. You cannot edit or switch to another component while in DISPLAY ROLLBACK.

To return to your editing session, you can enter RETURN on the command line or press the Return PF key (PF2 or PF14). The RETURN command returns you back to an editing session after a DISPLAY ROLLBACK.

When you sign on to CA Ideal after a system failure, you are returned to the last edit session at the time of the abend. At this point, you receive an error message. You can either view or edit the primary member or the backup member by issuing DISPLAY ROLLBACK.

This command has the following format:

```
DISPLAY ROLLBACK
```

Note: For report, plan, dataview, and panel definitions, all components of a report are affected if an abend occurs.

EXCLUDE Command

The EXCLUDE command locates all lines that do not contain a string of characters and suppresses the display of those lines in the display region.

The EXCLUDE command applies to certain components (see Components Used with the EXCLUDE Command). When editing a report definition, working data, or parameter definition, EXCLUDE applies only to the field name. You can use the ERROR operand to exclude statements the compiler marked as errors in a procedure definition.

Use the COLUMNS operand or the SET EDIT BOUNDS command to restrict the search to an inclusive range of columns. You can also limit the range of lines to search.

This command has the following format:

```
EXCLUDE [/string/] [LINE[S] start-line [end-line]]
        [ERROR ]
        [COLUMNS start-column [MAX      ]]
        [                [end-column]]
```

Operand Definitions

Operand	Description
/	Character that delimits the search string. You must use the same character consistently in a command. You can use any special character except the currently defined command delimiter (installed default is a semicolon (;)) or comment character (installed default is a colon (:), an asterisk (*), an at sign (@), a pound sign (#), or a dollar sign (\$)).
<i>string</i>	<p>A character or string of characters used in the search. If you do not specify a string, the string specified on the previous FIND, INCLUDE, EXCLUDE, FIRST, NEXT, PREVIOUS, or LAST command is used as a default value.</p> <p>The string cannot include the string delimiter character or the current command delimiter.</p>
ERROR	(Procedure definition only.) Excludes a line that the compiler marked as an error (using the MEL=YES operand of the COMPILE command and SET EDIT HIGHLIGHT ERRORS). You cannot specify column ranges.
<i>start-line</i>	<p>Line where the search for string begins. You can specify it as:</p> <p>num Sequence number of the start line. When you specify a single line with a sequence number, the keyword LINE is required.</p> <p>TOP Indicates the first line of the member. If you do not specify a start line, TOP is the default.</p> <p>CURSOR Indicates the position where the cursor is located.</p> <p>offset A position relative to the top line displayed:</p> <ul style="list-style-type: none"> * Top line displayed. *+ n Line n number of lines below the top line displayed. *- n Line n number of lines before the top line displayed.
<i>end-line</i>	<p>Line where the search for string ends. You can specify it as:</p> <p>BOTTOM (Default.) Last line of the member.</p> <p>num A sequence number.</p> <p>CURSOR Indicates the position where the cursor is located.</p> <p>offset A position relative to the top line displayed:</p> <ul style="list-style-type: none"> * Top line displayed. *+ n Line n number of lines below the top line displayed. *- n Line n number of lines before the top line displayed.
<i>start-column</i>	Column where the search begins.
<i>end-column</i>	Column where the search ends.

Operand	Description
MAX	Default. Furthest right-hand column.

Components Used with the EXCLUDE Command

The EXCLUDE command applies to the components listed below. Default column boundaries for each component are also listed.

	Column Edited	Column Boundaries
Member	Data area	1-72
Report		
Detail	Field name	1-25
Heading	Field name	1-56
Column	Field name	1-25
Program		
Procedure	Data area	1-72
Parameter	Field name	1-19
Working data	Field name	1-32
Dataview		
Field	Field name	1-19
Key (VSAM)	Field name	1-32
Plan		
DBRM	DBRM	1-8
Resource	Program name	1-8

Specifying lines or columns are optional. You can specify line and column clauses in either order. For example,

```
EXCL /DISPANEL/ LINES 16500 17500 COLUMNS 10 40
```

or,

```
EXCL /DISPANEL/ COLUMNS 10 40 LINES 16500 17500
```

Example

The EXCLUDE command in the following example searches for lines that do not contain the string FOR in columns 5 through 30.

```

=> EXCL /FOR/ COL 5 30

...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...
IDEAL: PROCEDURE DEFINITION PGM CURRENT (001) TEST          SYS: DEM  FILL-IN

Command...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== T O P =====
000100          FOR EACH EMPLOYEE
000200              WHERE STATE_ADDRESS = 'CA' OR 'IL'
000300              AND NUMBER > 125 AND < 150
000400              FOR PAYROLL
000500                  WHERE NUMBER = EMPLOYEE.NUMBER
000600                  AND NUMBER > 100 AND < 150
000700                  SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
000800                  SET STATE1 = 'CA'
000900                  SET STATE2 = 'IL'
001000                  PRODUCE ADRM1
001100                  ENDFOR
001200          ENDFOR
001300          FOR EACH EMPLOYEE
001400              ORDERED BY STATE_ADDRESS CITY_ADDRESS EMPLOYEE.NAME
001500              WHERE STATE_ADDRESS = 'CA' OR 'IL'

```

All lines that exclude the string are found and displayed in the region.

```

=>

...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...
IDEAL: PROCEDURE DEFINITION PGM CURRENT (001) TEST          SYS: DEM  FILL-IN
EXCL /FOR/
Command...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== T O P =====
000200              WHERE STATE_ADDRESS = 'CA' OR 'IL'
000300              AND NUMBER > 125 AND < 150
000500                  WHERE NUMBER = EMPLOYEE.NUMBER
000600                  AND NUMBER > 100 AND < 150
000700                  SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
000800                  SET STATE1 = 'CA'
000900                  SET STATE2 = 'IL'
001000                  PRODUCE ADRM1
001400          ORDERED BY STATE_ADDRESS CITY_ADDRESS EMPLOYEE.NAME
001500              WHERE STATE_ADDRESS = 'CA' OR 'IL'

```

FIND Command

The FIND command searches for lines that either contains a string or for lines that do not contain a string. You can use this command when displaying or editing certain components.

When editing a procedure definition, FIND can locate statements the compiler marked as errors. FIND applies only to the field name when editing a report definition, working data, or parameter definition.

You can direct FIND forward or backward. It can locate and display a specific number of lines. Use the COLUMNS operand or the SET EDIT BOUNDS command to restrict the FIND to an inclusive range of columns.

Once you find a line, you can specify FIND without the search string to locate the next line. You can specify any of the optional operands with subsequent FIND commands without specifying the search string. If you do not specify any optional operands with subsequent FIND commands, the operands specified on the previous FIND is used as default values.

This command has the following format:

```

    [1 ] [INCLUDE] [FORWARD ] [/string/]
FIND [n ] [EXCLUDE] [BACKWARD] [ERROR ]
    [ALL]
    [LINE[S] start-line [end-line]]
    [COLUMNS start-column [MAX      ]]
    [                [end-column]]

```

Operand Definitions

Operand	Description
1	Specifies the maximum number of lines to find. The default is 1.
<i>n</i>	Specifying 1 positions the first line at the top of the region with subsequent lines filling the region. Specify ALL to locate all lines.
ALL	If you specify ALL or any value greater than 1, only the lines containing the string display. Use scrolling commands or function keys to view all occurrences if the display exceeds the size of the region.
INCLUDE	Default. Searches for lines that contain the specified string.
EXCLUDE	Searches for lines that do not contain the specified string.
FORWARD	Default. Searches beginning with the first line and processing through subsequent lines.

Operand	Description
BACKWARD	Searches beginning with the last line and processing backward.
/	Character that delimits the search string. You must use the same character consistently in a command. You can use any special character except the currently defined command delimiter (installed default is a semicolon (;)) or comment character (installed default is a colon (:)), an asterisk (*), an at sign (@), a pound sign (#), or a dollar sign (\$).
<i>string</i>	<p>A character or string of characters used in the search. If you use the INCLUDE option, FIND searches for lines that contain this string. If you use the EXCLUDE operand, the command locates lines that do not contain this string. If you do not specify a string, the string specified on the previous FIND, INCLUDE, EXCLUDE, FIRST, NEXT, LAST, or PREVIOUS command is used as a default value.</p> <p>The string cannot include the string delimiter character or the current command delimiter.</p>
ERROR	(Procedure definition only.) Finds a line that the compiler marked as an error (using the MEL=YES operand of the COMPILE command and SET EDIT HIGHLIGHT ERRORS). You cannot specify column ranges.
<i>start-line</i>	<p>Line where the search for string-a begins. You can specify it as:</p> <p>num Sequence number of the start line. When you specify a single line with a sequence number, the keyword LINE is required.</p> <p>TOP Indicates the first line of the member. If you do not specify a start line, TOP is the default.</p> <p>CURSOR Indicates the position where the cursor is located.</p> <p>offset Position relative to the top line displayed:</p> <ul style="list-style-type: none"> * Top line displayed. *+ n Line n number of lines below the top line displayed. *- n Line n number of lines before the top line displayed.
<i>end-line</i>	<p>Line where the search for string-a ends. You can specify it as:</p> <p>BOTTOM Default. Last line of the member.</p> <p>num Sequence number.</p> <p>CURSOR Indicates the position where the cursor is located.</p> <p>offset Position relative to the top line displayed:</p> <ul style="list-style-type: none"> * Top line displayed. *+ n Line n number of lines below the top line displayed. *- n Line n number of lines before the top line displayed.

Operand	Description
<i>start-column</i>	Column where the search begins.
<i>end-column</i>	Column where the search ends.
<i>MAX</i>	Default. Farthest right-hand column.

Components Used with the FIND Command

The FIND command applies to the components listed below. Default column boundaries for each component are also listed.

	Column Edited	Column Boundaries
Member	Data area	1-72
Report		
Detail	Field name	1-25
Heading	Field name	1-56
Column	Field name	1-56
Program		
Procedure	Data area	1-72
Parameter	Field name	1-19
Working data	Field name	1-19
Dataview		
Field	Field name	1-19
Key (VSAM)	Field name	1-32
Plan		
DBRM	DBRM	1-8
Resource	Program name	1-8

Specifying lines or columns are optional. You can specify line and column clauses in either order. For example,

```
FIND /DISPANEL/ LINES 16500 17500 COLUMNS 10 40
```

or,

```
FIND /DISPANEL/ COLUMNS 10 40 LINES 16500 17500
```

If you use the *n* operand to search for all occurrences or any number of occurrences greater than 1, FIND displays only those lines containing the string. You can use the following commands with this display:

FIND, FIRST, NEXT, LAST, or PREVIOUS (where *n* > 1)
 IGNORE
 SCROLL

The use of any command other than those listed above replaces this display with a display showing the lines of the data in sequence.

Example

The FIND command in this example is looking for the string DISPANEL.NUMBER in a specific range.

```

=> FIND /DISPANEL.NUMBER/ LINES 16800 17500
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION PGM CURRENT (001) TEST      SYS: DEM  FILL-IN
.....1.....2.....3.....4.....5.....6.....7..Command
===== T O P =====
<<PROCESS_PANEL>> PROC                                016500
TRANSMIT_DISPANEL                                  016600
  SELECT                                           016700
    WHEN $ENTER-KEY                                016800
      FOR EMPLOYEE                                016900
        WHERE EMPLOYEE.NUMBER = DISPANEL.NUMBER    017000
        MOVE DISPANEL TO EMPLOYEE BY NAME          017100
        WHEN NONE                                  017200
        SET DISPANEL.MSG = 'EMPLOYEE DELETED'      017300
        SET ATTR HIGHLIGHT TEMP ON DISPANEL.NUMBER 017400
        ENDFOR                                     017500
    
```

The line with the string is found and positioned at the top of the region, as shown below, with subsequent lines filling the region. The find operation is then continued by specifying the command FIND with no string or by pressing F9/21 (FIND while in editing mode). The same LINE and COLUMN range is also assumed.

```

=>FIND
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION PGM CURRENT (001) TEST      SYS: DEM  FILL-IN
.....1.....2.....3.....4.....5.....6.....7..Command
===== T O P =====
    WHERE EMPLOYEE.NUMBER = DISPANEL.NUMBER        017000
    MOVE DISPANEL TO EMPLOYEE BY NAME              017100
    WHEN NONE                                       017200
    SET DISPANEL.MSG = 'EMPLOYEE DELETED'          017300
    SET ATTR HIGHLIGHT TEMP ON DISPANEL.NUMBER     017400
    ENDFOR                                         017500
    
```


The find operation continues and the line containing the next occurrence of the string originally specified in the FIND command is positioned to the top of the region.

```

.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION  PGM CURRENT (001) TEST          SYS: DEM  FILL-IN
.....1.....2.....3.....4.....5.....6.....7..Command
===== T O P =====
      SET ATTR HIGHLIGHT TEMP ON DISPANEL.NUMBER           017400
      ENDFOR                                                017500
    
```

The FIND command in the following example uses the operands ALL and EXCLUDE to search for all lines that do not contain the string EMPLOYEE.

```

=> FIND ALL EXCLUDE /EMPLOYEE/
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION PGM CRTEST (001) TEST      SYS: DEM  FILL-IN

.....1.....2.....3.....4.....5.....6.....7..Command
===== T O P =====
SELECT                                                    000100
WHEN $ENTER-KEY                                         000200
  FOR EMPLOYEE                                           000300
  WHERE EMPLOYEE.NUMBER = DISPANEL.NUMBER                000400
  MOVE DISPANEL TO EMPLOYEE BY NAME                     000500
  WHEN NONE                                              000600
  SET DISPANEL.MSG = 'EMPLOYEE DELETED'                 000700
  SET ATTR HIGHLIGHT TEMP ON DISPANEL.NUMBER            000800
  ENDFOR                                                 000900
  SET NEXT_PANEL = 'DISPANEL'                            001000
WHEN $PF11                                               001100
  SET NEXT_PANEL = 'EMPMENU'                            001200
WHEN $PF12                                               001300
  DO TERMINATE_SES                                      001400
WHEN $ENTER-KEY                                         001500
  FOR EMPLOYEE                                           001600
  WHERE EMPLOYEE.NUMBER = DISPANEL.NUMBER               001700
  
```

FIND displays all lines that do not contain the string. The status line shows the compare rule EXCL and the search string /EMPLOYEE/.

```

=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION PGM CRTEST (001) TEST      SYS: $ID  FILL-IN
EXCL /EMPLOYEE/
.....1.....2.....3.....4.....5.....6.....7..Command
SELECT                                                    000100
WHEN $ENTER-KEY                                         000200
  WHEN NONE                                              000600
  SET ATTR HIGHLIGHT TEMP ON DISPANEL.NUMBER            000800
  ENDFOR                                                 000900
  SET NEXT_PANEL = 'DISPANEL'                            001000
WHEN $PF11                                               001100
  SET NEXT_PANEL = 'EMPMENU'                            001200
WHEN $PF12                                               001300
  DO TERMINATE_SES                                      001400
WHEN $ENTER-KEY                                         001500
  
```

FIRST Command

The FIRST command locates the first occurrence of a string of characters. The search begins with the first line and processes forward through the last line.

You can use this command with certain components (see Notes). FIRST can locate statements the compiler marked as errors in a procedure definition. When editing a report definition, working data, or parameter definition, FIRST applies only to the field name.

Use the COLUMNS operand or the SET EDIT BOUNDS command to restrict the search to an inclusive range of columns. You can use FIRST to locate a specific number of occurrences.

This command has the following format:

```

[1 ] [/string/]
FIRST [n ] [ERROR ]
      [ALL]
      [COLUMNS start-column [MAX      ]]
      [                [end-column]]

```

Operand Definitions

Operand	Description
1 <i>n</i> ALL	Specifies the maximum number of occurrences of the string to find. The default is 1. Specifying 1 positions the first line containing the string to the top of the region with subsequent lines filling the region. Use ALL to locate all occurrences of the string. If you specify ALL or any value greater than 1, only the lines containing the string display. Use scrolling commands or function keys to view all occurrences if the display exceeds the size of the region.
/	Character that delimits the search string. You must use the same character consistently in a command. You can use any special character except the currently defined command delimiter (installed default is a semicolon (;)) or comment character (installed default is a colon (:)), an asterisk (*), an at sign (@), a pound sign (#), or a dollar sign (\$).
<i>string</i>	Character or string of characters to find. If you do not specify a string, the string specified on the previous FIND, INCLUDE, EXCLUDE, FIRST, NEXT, PREVIOUS, or LAST command is used as a default value. The string cannot include the string delimiter character or the current command delimiter.

Operand	Description
ERROR	(Procedure definition only.) Locates a line that the compiler marked as an error (using the MEL=YES operand of the COMPILE command and SET EDIT HIGHLIGHT ERRORS). You cannot specify column ranges.
<i>start-column</i>	Column where the search for the string begins.
<i>end-column</i>	Column where the search for the string ends.
MAX	Default. Farthest right-hand column.

Components Used with the FIRST Command

The FIRST command applies to the following components. Default column boundaries for each component are also listed.

	Column Edited	Column Boundaries
Member	Data area	1-72
Report		
Detail	Field name	1-25
Heading	Field name	1-56
Column	Field name	1-25
Program		
Procedure	Data area	1-72
Parameter	Field name	1-19
Working data	Field name	1-19
Dataview		
Field	Field name	1-19
Key (VSAM)	Field name	1-32
Plan		
DBRM	DBRM	1-8
Resource	Program name	1-8

If you use the n operand to search for all occurrences or any number of occurrences greater than 1, FIRST displays only those lines containing the string. You can use the following commands without changing this display.

```
FIND, FIRST, NEXT, LAST, PREVIOUS (where  $n > 1$ )
IGNORE
SCROLL
```

The use of any command other than those listed above replaces this display with a display showing the lines of data in sequence.

Example

The FIRST command searches for the first three occurrences of the string NUMBER.

```
=> FIRST 3 /NUMBER/
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..Command
===== T O P =====
SELECT                                                    000100
WHEN $ENTER-KEY                                         000200
  FOR EMPLOYEE                                           000300
  WHERE EMPLOYEE.NUMBER = DISPLANEL.NUMBER              000400
  MOVE DISPANEL TO EMPLOYEE BY NAME                    000500
  WHEN NONE                                              000600
  SET DISPANEL.MSG = 'EMPLOYEE DELETED'                000700
  SET ATTR HIGHLIGHT TEMP ON DISPANEL.NUMBER          000800
ENDFOR                                                  000900
  SET NEXT_PANEL = 'DISPANEL'                          001000
WHEN $PF11                                              001100
  SET NEXT_PANEL = 'EMPMENU'                          001200
WHEN $PF12                                              001300
  DO TERMINATE_SES                                     001400
WHEN $ENTER-KEY                                         001500
  FOR EMPLOYEE                                           001600
  WHERE EMPLOYEE.NUMBER = DISPLANEL.NUMBER            001700
```

The lines that contain the string are displayed.

```
=>
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN
INCL /NUMBER/
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..Command
===== T O P =====
  WHERE EMPLOYEE.NUMBER = DISPLANEL.NUMBER              000400
  SET ATTR HIGHLIGHT TEMP ON DISPANEL.NUMBER          000800
  WHERE EMPLOYEE.NUMBER = DISPLANEL.NUMBER            001700
```

IGNORE Line Command

The IGNORE line command refreshes only the line containing it. The contents of the line are restored as they were at the start of the current transaction. Any change made to the data on that line is not applied.

This command has the following format:

IGNORE

Example

In the following example, changes are made to the data in the region in the form of comments being added. The IGNORE line command is then entered on a line where a comment was typed.

```

=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION PGM CRTEST (001) TEST          SYS: $ID FILL-IN

.....1.....2.....3.....4.....5.....6.....7..Command
===== T O P =====
SELECT              :INCLUDE ADDITIONAL PF KEY              000100
WHEN $ENTER-KEY    :SELECTIONS                               000200
FOR EMPLOYEE      :AND THE CLEAR KEY                        IGNORE
WHERE EMPLOYEE.NUMBER = DISPLANEL.NUMBER                    000400
MOVE DISPANEL TO EMPLOYEE BY NAME                           000500
WHEN NONE          000600
SET DISPANEL.MSG = 'EMPLOYEE DELETED'                       000700
SET ATTR HIGHLIGHT TEMP ON DISPANEL.NUMBER                  000800
ENDFOR                                                      000900
    
```

The following example shows that the comment entered on the line containing the IGNORE line command was not applied.

```

=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION PGM CRTEST (001) TEST          SYS: $ID FILL-IN

.....1.....2.....3.....4.....5.....6.....7..Command
===== T O P =====
SELECT              :INCLUDE ADDITIONAL PF KEY              000100
WHEN $ENTER-KEY    :SELECTIONS                               000200
FOR EMPLOYEE      000300
WHERE EMPLOYEE.NUMBER = DISPLANEL.NUMBER                    000400
MOVE DISPANEL TO EMPLOYEE BY NAME                           000500
WHEN NONE          000600
SET DISPANEL.MSG = 'EMPLOYEE DELETED'                       000700
SET ATTR HIGHLIGHT TEMP ON DISPANEL.NUMBER                  000800
ENDFOR                                                      000900
    
```

INCLUDE Command

The INCLUDE command locates all lines that contain a string of characters and places those lines in the display region. The command applies to certain components (see Notes). When editing a report definition, working data, or parameter definition, INCLUDE applies only to the field name. You can use INCLUDE to locate statements the compiler marked as errors in a procedure definition.

Use the COLUMNS operand or the SET EDIT BOUNDS command to restrict the search to an inclusive range of columns. You can also limit the range of lines to search.

This command has the following format:

```
INCLUDE [/string/] [LINE[S] start-line [end-line]]
        [ERROR      ]
        [COLUMNS start-column [MAX      ]]
        [              [end-column]]
```

Operand Definitions

Operand	Description
/	The character that delimits the search string. You must use the same character consistently in a command. You can use any special character except the currently defined command delimiter (installed default is a semicolon (;)) or comment character (installed default is a colon (:)), an asterisk (*), an at sign (@), a pound sign (#), or a dollar sign (\$).
string	Character or string of characters to find. If you do not specify a string, the string specified on the previous FIND, INCLUDE, EXCLUDE, FIRST, NEXT, PREVIOUS, or LAST command is used as a default value. The string cannot include the string delimiter character or the current command delimiter.

ERROR (Procedure definition only.) Locates a line that the compiler marked as an error (using the MEL=YES operand of the COMPILE command and SET EDIT HIGHLIGHT ERRORS). You cannot specify column ranges.

start-line Line where the search for string begins. You can specify it as:

num Sequence number of the start line. When you specify a single line with a sequence number, the keyword LINE is required.

TOP Indicates the first line of the member. If you do not specify a start line, TOP is the default.

CURSOR Indicates the position where the cursor is located.

offset Position relative to the top line displayed:

* Top line displayed.

*+n Line n number of lines below the top line displayed.

*-n Line n number of lines before the top line displayed.

end-line Line where the search for string ends. You can specify it as:

BOTTOM Default. The last line of the member.

num Sequence number.

CURSOR Indicates the position where the cursor is located.

offset Position relative to the top line displayed:

* Top line displayed.

*+n Line n number of lines below the top line displayed.

*-n Line n number of lines before the top line displayed.

start-column Column where the search for the string begins.

end-column Column where the search for the string ends.

MAX Default. Farthest right-hand column.

The INCLUDE command applies to the components listed next. Default column boundaries for each component are also shown.

	Column Edited	Column Boundaries
Member	Data area	1-72
Report		
Detail	Field name	1-25

	Column Edited	Column Boundaries
Heading	Field name	1-56
Column	Field name	1-25
Program		
Procedure	Data area	1-72
Parameter	Field name	1-19
Working data	Field name	1-19
Dataview		
Field	Field name	1-19
Key (VSAM)	Field name	1-32
Plan		
DBRM	DBRM	1-8
Resource	Program name	1-8

Specifying lines or columns is optional. You can specify line and column clauses in either order. For example,

```
INCL /DISPANEL/ LINES 16500 17500 COLUMNS 10 40
```

or,

```
INCL /DISPANEL/ COLUMNS 10 40 LINES 16500 17500
```

Example

In the following example, INCLUDE searches for the string NUMBER, from the cursor position (line 300) through the last line of the member.

```

=> INCL /NUMBER/ CURSOR BOTTOM
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN

Command....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
===== T O P =====
000100          FOR EACH EMPLOYEE
000200              WHERE STATE_ADDRESS = 'CA' OR 'IL'
000300              AND NUMBER > 125 AND < 150
000400              FOR PAYROLL
000500                  WHERE NUMBER = EMPLOYEE.NUMBER
000600                  AND NUMBER > 100 AND < 150
000700                  SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
000800                  SET STATE1 = 'CA'
000900                  SET STATE2 = 'IL'
001000              PRODUCE CARM1
001100              ENDFOR
001200          ENDFOR
001300          FOR EACH EMPLOYEE
001400              ORDERED BY STATE_ADDRESS CITY_ADDRESS EMPLOYEE.NAME
001500              WHERE STATE_ADDRESS = 'CA' OR 'IL'
001600              AND NUMBER > 125 AND < 150
    
```

INCLUDE displays all lines in this range containing the string.

```

=>
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN
INCL /NUMBER/
Command....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
000300              AND NUMBER > 125 AND < 150
000500              WHERE NUMBER = EMPLOYEE.NUMBER
000600              AND NUMBER > 100 AND < 150
001600              AND NUMBER > 125 AND < 150
    
```

INPUT Command

The INPUT command inserts null lines (open a window) into the data based on the cursor position.

This command has the following format:

```

[[CONTEXT] n]
INPUT [WINDOW m ]
    
```

[CONTEXT] n

Indicates the number of required context lines (lines retained on the screen for reference). *n* lines display at the bottom of the screen if the cursor remains in the command area; or *n* lines appear at the top and bottom of the screen if the cursor is placed in the region. The rest of the screen is opened up for input.

You can set the default number of context lines with the SET EDIT CONTEXT command.

WINDOW m

Indicates the number of null lines to insert. The window is opened at the top of the screen if the cursor remains in the command area or following the line with the cursor if the cursor is placed in the region.

When possible, CA Ideal scrolls the window forward or backward to center the window in the region. When the value of *m* exceeds the number of lines left in the region, minus context lines, the maximum number of lines that can fit between the context lines is inserted.

Example

In the following example, a window 5 lines long is requested in the region starting below the line with the cursor (900).

```

=> INPUT WINDOW 5
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION PGM CRTEST (001) TEST          SYS: $ID FILL-IN

Command.....1.....2.....3.....4.....5.....6.....7..
===== T O P =====
000100          FOR EACH EMPLOYEE
000200              WHERE STATE_ADDRESS = 'CA' OR 'IL'
000300              AND NUMBER > 125 AND < 150
000400              FOR PAYROLL
000500                  WHERE NUMBER = EMPLOYEE.NUMBER
000600                  AND NUMBER > 100 AND < 150
000700                  SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
000800                  SET STATE1 = 'CA'
000900                  SET STATE2 = 'IL'
001000              PRODUCE CARM1
001100              ENDFOR
001200          ENDFOR
001300          FOR EACH EMPLOYEE
001400              ORDERED BY STATE_ADDRESS CITY_ADDRESS EMPLOYEE.NAME
001500              WHERE STATE_ADDRESS = 'CA' OR 'IL'
001600              AND NUMBER > 125 AND < 150

```

After applying the data, five null lines are inserted as shown in the next example.

```

=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION PGM CRTEST (001) TEST          SYS: $ID FILL-IN

Command.....1.....2.....3.....4.....5.....6.....7..
===== T O P =====
000100          FOR EACH EMPLOYEE
000200              WHERE STATE_ADDRESS = 'CA' OR 'IL'
000300              AND NUMBER > 125 AND < 150
000400              FOR PAYROLL
000500                  WHERE NUMBER = EMPLOYEE.NUMBER
000600                  AND NUMBER > 100 AND < 150
000700                  SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
000800                  SET STATE1 = 'CA'
000900                  SET STATE2 = 'IL'
.....
.....
.....
.....
.....
001000              PRODUCE CARM1
001100              ENDFOR
001200          ENDFOR
  
```

Input Line Command

The input line command opens a window of null lines for insertion of data. You can specify how many lines to insert.

This command has the following format:

```
[n]I[n]
[n]IB[n]
```

The SET EDIT MULTIPLIER command determines whether *n* is placed on the left or right.

I Specified on the line after which null lines are inserted.

IB Specified on the line before which null lines are inserted.

n Optional. The number of null lines (1 through 100) to insert. If you do not specify *n*, one null line is inserted.

If the value of *n* exceeds the number of lines remaining in the region, the maximum number of lines that can fit are inserted.

Any unused null lines are removed when the data is applied (when you press the Enter key or any function key).

Sequence numbers do not appear on blank lines inserted in a window until data is applied. The inserted lines are assigned sequence numbers in increments of one starting from the previous existing line. When the number of inserted lines exceeds the number of integers between the sequence numbers of the existing lines, the sequence numbers on the lines following the insertion are changed.

To renumber sequence numbers refer to the RENUMBER primary editing command.

Example

The following examples illustrate the use of the input line command and the numbering of inserted lines.

In the following example, the input line command, 5I, is entered in line 800.

```
=>
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN

Command.....1....+....2....+....3....+....4....+....5....+....6....+....7..
===== T O P =====
000100          FOR EACH EMPLOYEE
000200              WHERE STATE_ADDRESS = 'CA' OR 'IL'
000300              AND NUMBER > 125 AND < 150
000400              FOR PAYROLL
000500                  WHERE NUMBER = EMPLOYEE.NUMBER
000600                  AND NUMBER > 100 AND < 150
000700                  SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
5I0800                  SET STATE1 = 'CA'
000900                  SET STATE2 = 'IL'
001000              PRODUCE CARM1
001100              ENDFOR
001200          ENDFOR
```

After you press the Enter key, five null lines are inserted as shown in the next example. Neither the null lines nor the newly added lines have sequence numbers.

```
=>
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN

Command....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
=====  =====  T O P  =====
000100          FOR EACH EMPLOYEE
000200              WHERE STATE_ADDRESS = 'CA' OR 'IL'
000300              AND NUMBER > 125 AND < 150
000400              FOR PAYROLL
000500                  WHERE NUMBER = EMPLOYEE.NUMBER
000600                  AND NUMBER > 100 AND < 150
000700                  SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
000800                  SET STATE1 = 'CA'
.....                  SET STATE3 = 'VT'
.....                  SET STATE4 = 'ME'
.....
.....
.....
.....
000900                  SET STATE2 = 'IL'
001000              PRODUCE CARM1
001100              ENDFOR
001200          ENDFOR
```

After pressing the Enter key, the unused null lines are removed and the new lines are inserted with sequence numbers.

```
=>
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN

Command....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
=====  =====  T O P  =====
000100          FOR EACH EMPLOYEE
000200              WHERE STATE_ADDRESS = 'CA' OR 'IL'
000300              AND NUMBER > 125 AND < 150
000400              FOR PAYROLL
000500                  WHERE NUMBER = EMPLOYEE.NUMBER
000600                  AND NUMBER > 100 AND < 150
000700                  SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
000800                  SET STATE1 = 'CA'
000801                  SET STATE3 = 'VT'
000802                  SET STATE4 = 'ME'
000900                  SET STATE2 = 'IL'
001000              PRODUCE CARM1
001100              ENDFOR
001200          ENDFOR
```

In the following example, the input-before line command, IB2, is entered in line 1300.

```

=>
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION  PGM NEWTEST (001) TEST          SYS: $ID  FILL-IN

Command....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
=====  T O P  =====
000100          FOR EACH EMPLOYEE
000200              WHERE STATE_ADDRESS = 'CA' OR 'IL'
000300              AND NUMBER > 125 AND < 150
000400              FOR PAYROLL
000500                  WHERE NUMBER = EMPLOYEE.NUMBER
000600                  AND NUMBER > 100 AND < 150
000700                  SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
000800                  SET STATE1 = 'CA'
000801                  SET STATE3 = 'MO'
000900                  SET STATE2 = 'IL'
001000                  PRODUCE CARM1
001100                  ENDFOR
001200          ENDFOR
IB2300          FOR EACH EMPLOYEE
001400              ORDERED BY STATE_ADDRESS CITY_ADDRESS EMPLOYEE.NAME
001500              WHERE STATE_ADDRESS = 'CA' OR 'IL'
001600              AND NUMBER > 125 AND < 150

```

After you press the Enter key, two null lines are inserted as shown in the next example. The newly added lines do not have sequence numbers.

```

=>
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION  PGM NEWTEST (001) TEST          SYS: $ID  FILL-IN

Command....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
=====  =====  T O P  =====
000100          FOR EACH EMPLOYEE
000200              WHERE STATE_ADDRESS = 'CA' OR 'IL'
000300              AND NUMBER > 125 AND < 150
000400              FOR PAYROLL
000500                  WHERE NUMBER = EMPLOYEE.NUMBER
000600                  AND NUMBER > 100 AND < 150
000700                  SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
000800                  SET STATE1 = 'CA'
000801                  SET STATE3 = 'MO'
000900                  SET STATE2 = 'IL'
001000                  PRODUCE CARM1
001100                  ENDFOR
001200          ENDFOR
.....          WHEN $PF11
.....          SET NEXT_PANEL = 'EMPMENU'
001300          FOR EACH EMPLOYEE
001400              ORDERED BY STATE_ADDRESS CITY_ADDRESS EMPLOYEE.NAME
001500              WHERE STATE_ADDRESS = 'CA' OR 'IL'
001600              AND NUMBER > 125 AND < 150
    
```

After pressing the Enter key, the new lines are inserted with sequence numbers.

```

=>
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION  PGM NEWTEST (001) TEST          SYS: $ID  FILL-IN

Command....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
=====  =====  T O P  =====
000100          FOR EACH EMPLOYEE
000200              WHERE STATE_ADDRESS = 'CA' OR 'IL'
000300              AND NUMBER > 125 AND < 150
000400              FOR PAYROLL
000500                  WHERE NUMBER = EMPLOYEE.NUMBER
000600                  AND NUMBER > 100 AND < 150
000700                  SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
000800                  SET STATE1 = 'CA'
000801                  SET STATE3 = 'MO'
000900                  SET STATE2 = 'IL'
001000                  PRODUCE CARM1
001100                  ENDFOR
001200          ENDFOR
001201          WHEN $PF11
001202              SET NEXT_PANEL = 'EMPMENU'
001300          FOR EACH EMPLOYEE
001400              ORDERED BY STATE_ADDRESS CITY_ADDRESS EMPLOYEE.NAME
001500              WHERE STATE_ADDRESS = 'CA' OR 'IL'
001600              AND NUMBER > 125 AND < 150
    
```


LAST Command

The LAST command locates the last occurrence of a string of characters. The search begins with the last line and processes backward. The LAST command applies to certain components (see Notes). When editing a report definition, working data, or parameter definition, LAST applies only to the field name.

Use the COLUMNS operand or the SET EDIT BOUNDS command to restrict the search to an inclusive range of columns. LAST can also locate a specific number of occurrences. You can use the ERROR operand to locate statements marked as errors by the compiler in a procedure definition.

This command has the following format:

```

[1 ] [/string/]
LAST [n ] [ERROR ]
    [ALL]
    [COLUMNS start-column [MAX      ]]
    [                [end-column]]

```

1| n| ALL

The maximum number of occurrences of the string to find. The default is 1. Specifying 1 positions the last line containing the string at the top of the region with subsequent lines filling the region. Specify ALL to locate all occurrences of the string. If you specify ALL or any value greater than 1, only the lines containing the string display. Use scrolling commands or function keys to view all occurrences if the display exceeds the size of the region.

/

The character that delimits the search string. You must use the same character consistently in a command. You can use any special character except the currently defined command delimiter (installed default is a semicolon (;)) or comment character (installed default is a colon (:)), an asterisk (*), an at sign (@), a pound sign (#), or a dollar sign (\$).

string

A character or string of characters to find. If you do not specify a string, the string specified on the previous FIND, INCLUDE, EXCLUDE, FIRST, NEXT, PREVIOUS, or LAST command is used as a default value.

The string cannot include the string delimiter character or the current command delimiter.

ERROR (Procedure definition only)

Locates a line that the compiler marked as an error (using the MEL=YES operand of the COMPILE command and SET EDIT HIGHLIGHT ERRORS). You cannot specify column ranges.

start-column

The column where the search for the string begins.

end-column

The column where the search for the string ends.

MAX

Default. The furthest right-hand column.

Note: The LAST command applies to the components listed below. Default column boundaries for each component are also listed.

	Column Edited	Column Boundaries
Member	Data area	1-72
Report		
Detail	Field name	1-25
Heading	Field name	1-56
Column	Field name	1-25
Program		
Procedure	Data area	1-72
Parameter	Field name	1-19
Working data	Field name	1-19
Dataview		
Field	Field name	1-19
Key (VSAM)	Field name	1-32
Plan		
DBRM	DBRM	1-8
Resource	Program name	1-8

If you use the *n* operand to search for all occurrences or any number of occurrences greater than 1, LAST displays only those lines containing the string. You can use the following commands without changing this display:

```
FIND, FIRST, LAST, NEXT, or PREVIOUS (where n > 1)
IGNORE
SCROLL
```

The use of any command other than those listed above replaces this display with a display showing the lines of data in sequence.

MOVE Command

The MOVE command inserts the designated line or range of lines at a location marked as the destination of the move and deletes the line or lines at the original location. You can use the MOVE command with a CA Ideal member or a program's working data, parameter data, or procedure.

This command has the following format:

```
MOVE [LINES] start-line [end-line] target-line
```

start-line

The line where the move begins. You can specify it as:

nnn The sequence number of the start line.

TOP Indicates the first line of the entity. If you do not specify a start line, TOP is the default.

offset A position relative to the top line displayed:

* The top line displayed.

*+**n** The line *n* number of lines below the top line displayed.

*-**n** The line *n* number of lines before the top line displayed.

end-line

The line where the move ends. You can specify it as:

nnn Sequence number of the end line.

BOT Indicates the last line of the entity. If you do not specify an end line, BOTTOM is the default.

offset Position relative to the top line displayed:

* Top line displayed.

*+**n** Line *n* number of lines below the top line displayed.

*-**n** Line *n* number of lines before the top line displayed.

target-line

Line after which the moved lines are inserted. You can specify it as:

nnn Sequence number of the line.

TOP Indicates the first line of the entity.

BOT Indicates the last line of the entity.

offset Position relative to the top line displayed:

* The top line displayed.

*+**n** The line *n* number of lines below the top line displayed.

*-**n** The line *n* number of lines before the top line displayed.

Note: After the MOVE, new sequence numbers are assigned to the new locations. The command MOVE LINES TOP BOTTOM is prohibited.

Move Line Command

The move line command inserts the designated line or range of lines at a location marked as the destination of the move and deletes the line or lines at the original location. This line command also repeats the line or range of lines moved a specified number of times.

This command has the following format:

```
{M }  
{MM}  
  {A}  
[n]{B}[n]
```

M

Specifies a single line to move. You must specify a destination, indicated by A or B, with each M line command. The destination can occur before or after the M line command.

MM

Used in pairs to indicate both the start and end of the range of lines to move. You must specify a destination, indicated by the line command A or B, with each delimited range of lines. The destination can occur before or after the range. After specifying the start of a range with the MM line command, you can scroll the region forward or backward and perform other line commands except that you cannot specify other moves or copies while a move is pending..

A

Indicates that the destination for a line or a range of lines being moved is immediately after the line containing the A line command.

B

Indicates that the destination for a line or a range of lines being moved is immediately before the line containing the B line command.

n

Optionally, specifies the number of times (between 1 and 100) the line or range of lines is inserted at the specified destination. The SET EDIT MULTIPLIER command determines whether n is placed on the left or right. For example,

If multiplier is set with:	then you specify:
SET EDIT MULTIPLIER RIGHT	A7 or B7
SET EDIT MULTIPLIER LEFT	7A or 7B

You can mark only one destination (for a move or a copy) at any one time.

The inserted lines are assigned sequence numbers in increments of one starting from the previous existing line. When the number of inserted lines exceeds the number of integers between the sequence numbers of the existing lines, the sequence numbers on the lines following the insertion are changed. The following example illustrates moving a range of lines, 100 through 1601, before line 1606. Notice the PENDING message caused by scrolling the region forward after the start of the range is marked.

=> **SCROLL FORWARD**

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION PGM CRTEST (001) TEST          SYS: $ID FILL-IN
```

```
Command.....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
===== T O P =====
```

```
MM0100      FOR EACH EMPLOYEE
000200          WHERE STATE_ADDRESS = 'CA' OR 'IL'
000300          AND NUMBER > 125 AND < 150
000400          FOR PAYROLL
000500              WHERE NUMBER = EMPLOYEE.NUMBER
000600              AND NUMBER > 100 AND < 150
000700              SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
000800              SET STATE1 = 'CA'
000801              SET STATE3 = 'MO'
000900              SET STATE2 = 'IL'
001000          PRODUCE CARM1
001100          ENDFOR
001200      ENDFOR
001201      FOR EACH EMPLOYEE
001300      FOR EACH EMPLOYEE
001400          ORDERED BY STATE_ADDRESS CITY_ADDRESS EMPLOYEE.NAME
001500          WHERE STATE_ADDRESS = 'CA' OR 'IL'
```

```

=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN
PENDING: MM(100)
Command.....1.....2.....3.....4.....5.....6.....7..
001500          WHERE STATE_ADDRESS = 'CA' OR 'IL'
001600          AND NUMBER > 125 AND < 150
MM1601          WHEN $PF11
001602          SET NEXT_PANEL = 'EMPMENU'
001603          WHEN $PF12
001604          DO TERMINATE_SES
001605          WHEN OTHER
001606          SET DISPANEL.MSG = 'UNABLE TO INTERPRET KEY ENTERED'
001607          SET NEXT_PANEL = 'DISPANEL'
001608          ENDSSEL

```

After the data is applied, the original sequence numbers of the lines being moved are deleted and the lines are inserted in the new location with NEW sequence numbers.

```

=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN
Command.....1.....2.....3.....4.....5.....6.....7..
===== T O P =====
001602          SET NEXT_PANEL = 'EMPMENU'
001603          WHEN $PF12
001604          DO TERMINATE_SES
001605          WHEN OTHER
001606          FOR EACH EMPLOYEE
001607          WHERE STATE_ADDRESS = 'CA' OR 'IL'
001608          AND NUMBER > 125 AND < 150
001609          FOR PAYROLL
001610          WHERE NUMBER = EMPLOYEE.NUMBER
001611          AND NUMBER > 100 AND < 150
001612          SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
001613          SET STATE1 = 'CA'
001614          SET STATE3 = 'MO'
001615          SET STATE2 = 'IL'
001616          PRODUCE CARM1
001617          ENDFOR
001618          ENDFOR

```

NEXT Command

The NEXT command locates the next occurrence of a string of characters. The search begins with the second line of the display and processes forward.

The NEXT command applies to certain components (see Notes). When editing a report definition, working data, or parameter definition, NEXT applies only to the field name. You can use the ERROR operand to locate statements the compiler marked as errors in a procedure definition.

Use the COLUMNS operand or the SET EDIT BOUNDS command to restrict the search to an inclusive range of columns. NEXT can also locate a specific number of occurrences.

This command has the following format:

```

[1 ] [/string/]
NEXT [n ] [ERROR ]
[ALL]
[COLUMNS start-column [MAX ]]
[
[end-column]]

```

1| n| ALL

The maximum number of occurrences of the string to find. The default is 1. Specifying 1 positions the next line containing the string at the top of the region with subsequent lines filling the region. Specify ALL to locate all occurrences of the string. If you specify ALL or any value greater than 1, only the lines containing the string displays. Use scrolling commands or function keys to view all occurrences if the display exceeds the size of the region.

/

The character that delimits the search string. You must use the same character consistently in a command. You can use any special character except the currently defined command delimiter (installed default is a semicolon (;)) or comment character (installed default is a colon (:)), an asterisk (*), an at sign (@), a pound sign (#), or a dollar sign (\$).

string

A character or string of characters to find. If you do not specify a string, the string specified on the previous FIND, INCLUDE, EXCLUDE, FIRST, NEXT, PREVIOUS, or LAST command is used as a default value.

The string cannot include the string delimiter character or the current command delimiter.

ERROR (Procedure definition only)

Locates a line that the compiler marked as an error (using the MEL=YES operand of the COMPILE command and SET EDIT HIGHLIGHT ERRORS). You cannot specify column ranges.

start-column

The column where the search for the string begins.

end-column

The column where the search for the string ends.

MAX

Default. The furthest right-hand column.

The NEXT command applies to the components listed below. Default column boundaries for each component are also listed.

	Column Edited	Column Boundaries
Member	Data area	1-72
Report		
Detail	Field name	1-25
Heading	Field name	1-56
Column	Field name	1-25
Program		
Procedure	Data area	1-72
Parameter	Field name	1-19
Working data	Field name	1-19
Dataview		
Field	Field name	1-19
Key (VSAM)	Field name	1-32
Plan		
DBRM	DBRM	1-8
Resource	Program name	1-8

If you specify the n operand as ALL or any value greater than 1, only the lines that contain the string display. You can use the following commands on this display:

- FIND, FIRST, NEXT, PREVIOUS or LAST (where n > 1)
- IGNORE
- SCROLL

The use of any command other than those listed above replaces this display with a display showing the lines of data in sequence.

POSITION Command

The POSITION command positions a specific line at the top of the region.

This command has the following format:

POSITION *n*

n

For programs, members, and report definitions, *n* is the sequence number of the line to position at the top of the region.

For outputs and panel layouts, *n* is the absolute line number in that component. (1 is the first line, 2 is the second line, and so on).

For the panel field summary table, *n* is the field number; for example, POSITION 7 brings the row for field 7 to the top of the region in the layout portion and the entry for field 7 to the top of the table portion of the region.

Example

The following example positions line 1612 at the top of the region.

```

=> POSITION 1612
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION PGM CRTEST (001) TEST          SYS: $ID FILL-IN

Command....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
===== T O P =====
001602          SET NEXT_PANEL = 'EMPMENU'
001603          WHEN $PF12
001604              DO TERMINATE_SES
001605          WHEN OTHER
001606          FOR EACH EMPLOYEE
001607              WHERE STATE_ADDRESS = 'CA' OR 'IL'
001608              AND NUMBER > 125 AND < 150
001609          FOR PAYROLL
001610              WHERE NUMBER = EMPLOYEE.NUMBER
001611              AND NUMBER > 100 AND < 150
001612          SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
001613          SET STATE1 = 'CA'
001614          SET STATE3 = 'MO'
001615          SET STATE2 = 'IL'
001616          PRODUCE CARM1
001617          ENDFOR
001618          ENDFOR

```

Line 1612 is now at the top of the region with subsequent lines filling the region.

```
=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN

Command.....1.....2.....3.....4.....5.....6.....7..
001612          SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
001613          SET STATE1 = 'CA'
001614          SET STATE3 = 'MO'
001615          SET STATE2 = 'IL'
001616          PRODUCE CARM1
001617          ENDFOR
001618          ENDFOR
001619          FOR EACH EMPLOYEE
001620          FOR EACH EMPLOYEE
001621          ORDERED BY STATE_ADDRESS CITY_ADDRESS EMPLOYEE.NAME
001622          WHERE STATE_ADDRESS = 'CA' OR 'IL'
001623          AND NUMBER > 125 AND < 150
001624          WHEN $PF11
001625          SET DISPANEL.MSG = 'UNABLE TO INTERPRET KEY ENTERED'
001626          SET NEXT_PANEL = 'DISPANEL'
001627          ENDSSEL
```

PREVIOUS Command

The PREVIOUS command locates the previous occurrence of a string of characters. The search begins with the line before the current top line of the display region and processes backward.

The PREVIOUS command applies to certain components (see Notes). When editing a report definition, working data, or parameter definition, PREVIOUS applies only to the field name. You can use the ERROR operand to locate statements the compiler marked as an error in the procedure definition.

Use the COLUMNS operand or the SET EDIT BOUNDS command to restrict the search to an inclusive range of columns. You can also locate a specific number of occurrences.

This command has the following format:

```
PREVIOUS [1 ] [/string/]
          [n ] [ERROR ]
          [ALL]
          [COLUMNS start-column [MAX      ]]
          [                [end-column]]
```

1| n| ALL

The maximum number of occurrences of the string to find. The default is 1. Specifying 1 positions the previous occurrence of the string at the top of the region with subsequent lines filling the region. Specify ALL to locate all occurrences of the string. If you specify ALL or any value greater than 1, only the lines containing the string display. Use scrolling commands or function keys to view all occurrences if the display exceeds the size of the region.

/

The character that delimits the search string. You must use the same character consistently in a command. You can use any special character except the currently defined command delimiter (installed default is a semicolon (;)) or comment character (installed default is a colon (:)), an asterisk (*), an at sign (@), a pound sign (#), or a dollar sign (\$).

string

A character or string of characters to find. If you do not specify a string, the string specified on the previous FIND, INCLUDE, EXCLUDE, FIRST, NEXT, PREVIOUS, or LAST command is used as a default value.

The string cannot include the string delimiter character or the current command delimiter.

ERROR (Procedure definition only)

Locates a line that the compiler marked as an error (using the MEL=YES operand of the COMPILE command and SET EDIT HIGHLIGHT ERRORS). You cannot specify column ranges.

start-column

The column where the search for the string begins.

end-column

The column where the search for the string ends.

MAX

Default. The furthest right-hand column.

The PREVIOUS command applies to the components listed below. Default column boundaries for each component are also listed.

	Column Edited	Column Boundaries
Member	Data area	1-72
Report		
Detail	Field name	1-25
Heading	Field name	1-56
Column	Field name	1-25
Program		
Procedure	Data area	1-72
Parameter	Field name	1-19
Working data	Field name	1-19
Dataview		
Field	Field name	1-19
Key (VSAM)	Field name	1-32
Plan		
DBRM	DBRM	1-8
Resource	Program name	1-8

If you use the *n* operand and specify ALL or any value greater than 1, only the lines that contain the string display. You can use the following commands without changing this display:

```
FIND, FIRST, NEXT, PREVIOUS, or LAST (where n > 1)
IGNORE
SCROLL
```

The use of any command other than those listed above replaces this display with a display showing the lines of data in sequence.

RENUMBER Command

The RENUMBER command renumbers the sequence numbers of the current entity and sets the increment the sequence numbers use.

This command has the following format:

```
RENUMBER [[BY] n]
```

[BY] n

Specifies the increment used in the sequence numbers. The value of *n* can be any integer (that completely renumbers the entity). If no increment is specified, a default of 100 is used. However, if RENUMBER BY 10 is issued, then subsequent renumber commands, without specifying an increment, still use 10.

Example

The following example illustrates the RENUMBER command. First, a move is specified.

```
=>
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN

Command....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
=====  =====  T O P  =====
001602          SET NEXT_PANEL = 'EMPMENU'
001603          WHEN $PF12
001604              DO TERMINATE_SES
001605          WHEN OTHER
001606          FOR EACH EMPLOYEE
MM1607              WHERE STATE_ADDRESS = 'CA' OR 'IL'
001608              AND NUMBER > 125 AND < 150
001609              FOR PAYROLL
001610                  WHERE NUMBER = EMPLOYEE.NUMBER
001611                  AND NUMBER > 100 AND < 150
MM1612                  SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
001613                  SET STATE1 = 'CA'
001614                  SET STATE3 = 'MO'
001615                  SET STATE2 = 'IL'
B01616          PRODUCE CARM1
001617          ENDFOR
001618          ENDFOR
```

After the move, new sequence numbers are assigned to the new locations. The RENUMBER command is entered.

```

=> RENUMBER BY 100
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN

Command.....1.....2.....3.....4.....5.....6.....7..
===== T O P =====
001602          SET NEXT_PANEL = 'EMPMENU'
001603          WHEN $PF12
001604            DO TERMINATE_SES
001605          WHEN OTHER
001606            FOR EACH EMPLOYEE
001613              SET STATE1 = 'CA'
001614              SET STATE3 = 'MO'
001615              SET STATE2 = 'IL'
001616              WHERE STATE_ADDRESS = 'CA' OR 'IL'
001617              AND NUMBER > 125 AND < 150
001618              FOR PAYROLL
001619              WHERE NUMBER = EMPLOYEE.NUMBER
001620              AND NUMBER > 100 AND < 150
001621              SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
001622            PRODUCE CARM1
001623            ENDFOR
001624          ENDFOR
    
```

The following example illustrates the renumbered sequence numbers.

```

=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN

Command.....1.....2.....3.....4.....5.....6.....7..
===== T O P =====
000100          SET NEXT_PANEL = 'EMPMENU'
000200          WHEN $PF12
000300            DO TERMINATE_SES
000400          WHEN OTHER
000500            FOR EACH EMPLOYEE
000600              SET STATE1 = 'CA'
000700              SET STATE3 = 'MO'
000800              SET STATE2 = 'IL'
000900              WHERE STATE_ADDRESS = 'CA' OR 'IL'
001000              AND NUMBER > 125 AND < 150
001100              FOR PAYROLL
001200              WHERE NUMBER = EMPLOYEE.NUMBER
001300              AND NUMBER > 100 AND < 150
001400              SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
001500            PRODUCE CARM1
001600            ENDFOR
001700          ENDFOR
    
```

Repeat Line Command

The repeat line command duplicates a line or range of lines. Lines duplicated with the repeat line commands are inserted immediately after the line containing the repeat line command (R) or the ending repeat range line command (RR).

This command has the following format:

```
{R }
[n]{RR}[n]
```

R

Specified on the line to repeat.

RR

Used in pairs to indicate the start and end of a range of lines to repeat. The second occurrence of this command on the same or subsequent screen is interpreted as the end of the range.

After specifying the start of a range with the RR line command, you can scroll the region forward or backward and perform other line commands. Until the end of the range started by an RR line command is found, the PENDING message appears in the message line.

n

Optionally specifies the number of times (between 1 and 100) the line or range of lines is repeated. If you do not specify n, the line is repeated once. The SET EDIT MULTIPLIER command determines whether n is placed on the left or right. For example,

If multiplier is set with:	then you specify:
SET EDIT MULTIPLIER RIGHT	R7
SET EDIT MULTIPLIER LEFT	7R

When interpreting line commands that mark ranges of lines to repeat, the commands are matched into pairs from the top down, regardless of the order in which you entered the line commands.

The inserted lines are assigned sequence numbers in increments of one, starting from the previous existing line. When the number of inserted lines exceeds the number of integers between the sequence numbers of the existing lines, the sequence numbers on the lines following the insertion are changed. (See the following example.) To renumber the sequence numbers, refer to the RENUMBER primary editing command.

Example

The following example shows the use of the repeat line command. Line 100 contains a line command that repeats a line twice.

```
=>
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN

Command....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
===== T O P =====

2R0100          SET NEXT_PANEL = 'EMPMENU'
000200          WHEN $PF12
000300            DO TERMINATE_SES
000400          WHEN OTHER
000500            FOR EACH EMPLOYEE
000600              SET STATE1 = 'CA'
000700              SET STATE3 = 'MO'
000800              SET STATE2 = 'IL'
000900              WHERE STATE_ADDRESS = 'CA' OR 'IL'
001000              AND NUMBER > 125 AND < 150
001100              FOR PAYROLL
001200                WHERE NUMBER = EMPLOYEE.NUMBER
001300                AND NUMBER > 100 AND < 150
001400                SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
001500              PRODUCE CARM1
001600            ENDFOR
001700          ENDFOR
```

The following example shows the repeated lines with sequence numbers assigned (101 and 102).

```
=>
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN

Command....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
===== T O P =====

000100          SET NEXT_PANEL = 'EMPMENU'
000101          SET NEXT_PANEL = 'EMPMENU'
000102          SET NEXT_PANEL = 'EMPMENU'
000200          WHEN $PF12
000300            DO TERMINATE_SES
000400          WHEN OTHER
000500            FOR EACH EMPLOYEE
000600              SET STATE1 = 'CA'
000700              SET STATE3 = 'MO'
000800              SET STATE2 = 'IL'
000900              WHERE STATE_ADDRESS = 'CA' OR 'IL'
001000              AND NUMBER > 125 AND < 150
001100              FOR PAYROLL
001200                WHERE NUMBER = EMPLOYEE.NUMBER
001300                AND NUMBER > 100 AND < 150
001400                SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
001500              PRODUCE CARM1
001600            ENDFOR
```


In the following example, lines 1000 and 1400 contain line commands that repeat a range of lines.

```
=>
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID FILL-IN

Command....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
===== T O P =====
000100          SET NEXT_PANEL = 'EMPMENU'
000101          SET NEXT_PANEL = 'EMPMENU'
000200          WHEN $PF12
000300            DO TERMINATE_SES
000400          WHEN OTHER
000500            FOR EACH EMPLOYEE
000600              SET STATE1 = 'CA'
000700              SET STATE3 = 'MO'
000800              SET STATE2 = 'IL'
000900              WHERE STATE_ADDRESS = 'CA' OR 'IL'
RR1000          AND NUMBER > 125 AND < 150
001100          FOR PAYROLL
001200            WHERE NUMBER = EMPLOYEE.NUMBER
001300            AND NUMBER > 100 AND < 150
RR1400          SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
001500          PRODUCE CARM1
001600          ENDFOR
```

The following example shows the repeated lines inserted after line 1400 with sequence numbers assigned (1401 through 1405).

```
=>
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID FILL-IN

Command....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
===== T O P =====
000100          SET NEXT_PANEL = 'EMPMENU'
000101          SET NEXT_PANEL = 'EMPMENU'
000200          WHEN $PF12
000300            DO TERMINATE_SES
000400          WHEN OTHER
000500            FOR EACH EMPLOYEE
000600              SET STATE1 = 'CA'
000700              SET STATE3 = 'MO'
000800              SET STATE2 = 'IL'
000900              WHERE STATE_ADDRESS = 'CA' OR 'IL'
001000          AND NUMBER > 125 AND < 150
001100          FOR PAYROLL
001200            WHERE NUMBER = EMPLOYEE.NUMBER
001300            AND NUMBER > 100 AND < 150
001400          SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
001401          AND NUMBER > 125 AND < 150
001402          FOR PAYROLL
001403            WHERE NUMBER = EMPLOYEE.NUMBER
001404            AND NUMBER > 100 AND < 150
001405          SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
001500          PRODUCE CARM1
001600          ENDFOR
```

RESET Command

The RESET command resets any pending M, MM, C, CC, DD, RR, A, or B line commands and errors. This primary command, issued in the command area, performs the same function as the RESET line command.

This command has the following format:

```
RESET
```

RESET Line Command

The RESET line command resets any pending M, MM, C, CC, RR, A, or B line commands and errors.

This command has the following format:

```
RESET
```

You can enter the RESET line command on any line. It applies to all pending line commands.

Example

The following example illustrates how you enter the RESET line command in the sequence number and command field. A move is pending on lines 500 through 700.

```
=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION PGM CRTEST (001) TEST          SYS: $ID FILL-IN
PENDING: MM(500) MM(700)
Command.....1.....2.....3.....4.....5.....6.....7..
001402                FOR PAYROLL
001403                WHERE NUMBER = EMPLOYEE.NUMBER
001404                AND NUMBER > 100 AND < 150
001405                SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
001500                PRODUCE CARM1
001600                ENDFOR
001700                ENDFOR
RESET0              FOR EACH EMPLOYEE
001900                FOR EACH EMPLOYEE
002000                ORDERED BY STATE_ADDRESS CITY_ADDRESS EMPLOYEE.NAME
002100                WHERE STATE_ADDRESS = 'CA' OR 'IL'
002200                AND NUMBER > 125 AND < 150
002300                WHEN $PF11
002400                SET DISPANEL.MSG = 'UNABLE TO INTERPRET KEY ENTERED'
002500                SET NEXT_PANEL = 'DISPANEL'
002600                ENDSEL
===== B O T T O M =====
```

After the data is applied, the pending move command is deleted and the PENDING message ends.

```

=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN

Command.....1.....2.....3.....4.....5.....6.....7..
001402          FOR PAYROLL
001403          WHERE NUMBER = EMPLOYEE.NUMBER
001404          AND NUMBER > 100 AND < 150
001405          SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
001500          PRODUCE CARM1
001600          ENDFOR
001700          ENDFOR
001800          FOR EACH EMPLOYEE
001900          FOR EACH EMPLOYEE
002000          ORDERED BY STATE_ADDRESS CITY_ADDRESS EMPLOYEE.NAME
002100          WHERE STATE_ADDRESS = 'CA' OR 'IL'
002200          AND NUMBER > 125 AND < 150
002300          WHEN $PF11
002400          SET DISPANEL.MSG = 'UNABLE TO INTERPRET KEY ENTERED'
002500          SET NEXT_PANEL = 'DISPANEL'
002600          ENDSEL
===== B O T T O M =====

```

RESHOW Line Command

The RESHOW line command refreshes the region (that is, <<PROCESS-PANEL>> PROC) with the data as it originally appeared before data was entered for the current transaction. Changes made to the data on the screen during the current transaction are not applied. You can specify the RESHOW line command on any line in the region. It applies to the entire region.

This command has the following format:

```
RESHOW
```

Example

The following example illustrates changes made to data in the region and the restoration of the data in the region without the changes being applied. The current transaction begins with the following data in the region.

```

=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION  PGM CRPNL (001) TEST          SYS: $ID  FILL-IN

Command.....1.....2.....3.....4.....5.....6.....7..
===== T O P =====
000001 <<PROCESS_PANEL>> PROC
000002     TRANSMIT DISPANEL
000003     SELECT
000004     WHEN $PF11
000005         SET NEXT_PANEL = 'EMPMENU'
000006     WHEN $PF12
000007         DO TERMINATE_SES
000008     WHEN OTHER
000009         SET DISPANEL.MSG = 'UNABLE TO INTERPRET KEY ENTERED'
000010         SET NEXT_PANEL = 'DISPANEL'
000011     ENDSSEL
===== B O T T O M =====

```

The data is then changed on the screen (line 7) and delete line commands are specified. The RESHOW line command is then entered on line 1 (although you can specify the RESHOW command anywhere in the region).

```

=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION  PGM CRPNL (001) TEST          SYS: $ID  FILL-IN

Command.....1.....2.....3.....4.....5.....6.....7..
===== T O P =====
RESHOW <<PROCESS_PANEL>> PROC
000002     TRANSMIT DISPANEL
000003     SELECT
D00004     WHEN $PF11
D00005         SET NEXT_PANEL = 'EMPMENU'
000006     WHEN $PF12
000007         DO CANCEL-SESSION
000008     WHEN OTHER
000009         SET DISPANEL.MSG = 'UNABLE TO INTERPRET KEY ENTERED'
000010         SET NEXT_PANEL = 'DISPANEL'
000011     ENDSSEL
===== B O T T O M =====

```

The result is the data as it appeared at the start of the current transaction with no changes applied to the data and no lines deleted.

```

=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION  PGM CRPNL (001) TEST          SYS: $ID  FILL-IN

Command.....1.....2.....3.....4.....5.....6.....7..
===== T O P =====
000001 <<PROCESS_PANEL>> PROC
000002     TRANSMIT DISPANEL
000003     SELECT
000004     WHEN $PF11
000005         SET NEXT_PANEL = 'EMPMENU'
000006     WHEN $PF12
000007         DO TERMINATE_SES
000008     WHEN OTHER
000009         SET DISPANEL.MSG = 'UNABLE TO INTERPRET KEY ENTERED'
000010         SET NEXT_PANEL = 'DISPANEL'
000011     ENDSSEL
===== B O T T O M =====

```

ROLLBACK Command

The ROLLBACK command restores the contents of the component as of the last CHECKPOINT or as of the beginning of the edit session (the automatic checkpoint).

This command has the following format:

```
ROLLBACK
```

When you issue the ROLLBACK command while editing the working data, parameter data, or procedure of a program definition, only that component is restored. When issued while editing a report or panel definition, the entire definition is restored.

Scroll Command

The SCROLL command lets you view and edit data outside the screen presentation area. You can scroll vertically or horizontally.

Vertical Scrolling

The following form of the SCROLL command moves the window overlooking data forward or backward, according to the cursor or frame position, or by number of pages or lines. You can also move the window to the top or bottom of the data.

This command has the following format:

```

    {{FORWARD } [ CURSOR  ]}
    {{BACKWARD} [ FRAME   ]}
    {{+       } [n {PAGES} ]}
    {{-       } [ {LINES} ]}
[SCROLL] {
    {{TOP     }           }
    {{BOTTOM }           }

```

The keyword SCROLL is optional.

FORWARD

Scrolls forward either one frame (when used with the FRAME option) or until the line containing the cursor is positioned at the top of the region (when used with the CURSOR option).

BACKWARD

Scrolls backward either one frame (when used with the FRAME option) or until the line containing the cursor is positioned at the top of the region (when used with the CURSOR option).

- + equivalent to FORWARD.
- equivalent to BACKWARD.

CURSOR

Scrolls forward or backward until the line containing the cursor is at the bottom or top of the region, respectively. If SET SCROLL CURSOR is in effect, CURSOR is the default.

FRAME

Scrolls forward or backward by an entire frame, but repeats one context line at the top of the next frame from the current frame. If SET SCROLL FRAME is in effect, FRAME is the default.

n PAGES (For outputs in the output library only)

Scrolls forward or backward the specified number of report pages.

n LINES

Scrolls forward or backward the specified number of lines.

TOP

Positions the window so that the first line of data is at the top of the region.

BOTTOM

Positions the window so that the last line of the data is at the bottom of the region.

To use a + as SCROLL FORWARD or - as SCROLL BACKWARD, you must define other characters as the current defaults for the REPEAT and RESHOW commands.

If you specify TOP or BOTTOM, you cannot include any other option.

Example

```

SCROLL +
+
SCROLL -
-
SCROLL FOR CUR
SCR FOR 12 LINES
SCR BAC 7 PAGES
SCROLL TOP
TOP
SCROLL BOTTOM
BOTTOM

```

Horizontal Scrolling

The following SCROLL command scrolls the window overlooking an output in the output library or a panel layout in a horizontal direction. You cannot use options with column numbers with output members.

This command has the following format:

```

[ [CURSOR ] ]
[RIGHT [FRAME ] ]
[SCROLL] [LEFT [MAX ] ]
[ [nnn [COLUMNS]] ]
[COLUMN nnn ]

```

The keyword SCROLL is optional if you specify at least one other argument.

LEFT (Default)

Scrolls the window to the left (by cursor, frame, or column, or until the left margin).

RIGHT

Scrolls the window to the right (by cursor, frame, or column, or until the right margin).

CURSOR

Scrolls left or right until the column containing the cursor is at the left or right of the region, respectively. If SET SCROLL CURSOR is in effect, CURSOR is the default.

FRAME

Scrolls left or right by an entire frame, but repeats one context column at the left or right of the next frame. If SET SCROLL FRAME is in effect, FRAME is the default.

MAX

Specifies the left or right scroll as to the left or right edge of the defined panel.

***nnn* COLUMNS**

Specifies the number of columns, *nnn*, to scroll to the left or right from the current position.

COLUMN *nnn*

Specifies the absolute column *nnn* as the left margin. LEFT and RIGHT are optional with SCROLL COLUMN.

Scroll Line Command

The scroll line command scrolls a specific line to the top of the region with subsequent lines filling the region.

This command has the following format:

[+*n*]
*[-*n*]

* Positions the line to the top of the region.

+*n* The line *n* number of lines after the line containing the command.

-*n* The line *n* number of lines before the line containing the command. The value of *n* can be any number from 1 to 100.

Example

The following example shows the use of the scroll line command, an asterisk (*) in line 1100, to position line 1100 to the top of the region.

```

=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN

Command.....1.....2.....3.....4.....5.....6.....7..
===== T O P =====
000100          SET NEXT_PANEL = 'EMPMENU'
000200          SET NEXT_PANEL = 'EMPMENU'
000300          WHEN $PF12
000400            DO TERMINATE_SES
000500          WHEN OTHER
000600            FOR EACH EMPLOYEE
000700              SET STATE1 = 'CA'
000800              SET STATE3 = 'MO'
000900              SET STATE2 = 'IL'
001000              WHERE STATE_ADDRESS = 'CA' OR 'IL'
*01100              AND NUMBER > 125 AND < 150
001200              FOR PAYROLL
001300                WHERE NUMBER = EMPLOYEE.NUMBER
001400                AND NUMBER > 100 AND < 150
001500                SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
001600                AND NUMBER > 125 AND < 150
001700              FOR PAYROLL

```

Line 1100 is now at the top of the region with subsequent lines filling the region. To position line 200 to the top of the region, you can use the command *-9.

```

=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN

Command.....1.....2.....3.....4.....5.....6.....7..
*-9100          AND NUMBER > 125 AND < 150
001200          FOR PAYROLL
001300            WHERE NUMBER = EMPLOYEE.NUMBER
001400            AND NUMBER > 100 AND < 150
001500            SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
001600            AND NUMBER > 125 AND < 150
001700            FOR PAYROLL
001800            WHERE NUMBER = EMPLOYEE.NUMBER
001900            AND NUMBER > 100 AND < 150
002000            SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
002100            PRODUCE CARM1
002200            ENDFOR
002300          ENDFOR
002400          FOR EACH EMPLOYEE
002500          FOR EACH EMPLOYEE
002600            ORDERED BY STATE_ADDRESS CITY_ADDRESS EMPLOYEE.NAME
002700            WHERE STATE_ADDRESS = 'CA' OR 'IL'
002800            AND NUMBER > 125 AND < 150

```

Line 200 is now at the top of the region with subsequent lines filling the region.

```
=>
.....1.....2.....3.....4.....5.....6.....7.....
IDEAL: PROCEDURE DEFINITION  PGM CRTEST (001) TEST          SYS: $ID  FILL-IN

Command.....1.....2.....3.....4.....5.....6.....7..
000200          SET NEXT_PANEL = 'EMPMENU'
000300          WHEN $PF12
000400            DO TERMINATE_SES
000500          WHEN OTHER
000600            FOR EACH EMPLOYEE
000700              SET STATE1 = 'CA'
000800              SET STATE3 = 'MO'
000900              SET STATE2 = 'IL'
001000              WHERE STATE_ADDRESS = 'CA' OR 'IL'
001100              AND NUMBER > 125 AND < 150
001200              FOR PAYROLL
001300                WHERE NUMBER = EMPLOYEE.NUMBER
001400                AND NUMBER > 100 AND < 150
001500                SET SALARY = YTD_WAGES + YTD_COMMISSION - YTD_TAX
001600              AND NUMBER > 125 AND < 150
001700              FOR PAYROLL
001800                WHERE NUMBER = EMPLOYEE.NUMBER
001900                AND NUMBER > 100 AND < 150
```

SHIFT Command

The SHIFT command shifts a line or range of lines by one or more columns to the right or to the left. The SHIFT command only applies to programs (procedure definition) and data members. You can restrict SHIFT to an inclusive range of lines and columns using the format below or using the SET EDIT BOUNDS command.

This command has the following format:

```
{RIGHT}
SHIFT {LEFT } nnn [LINE[S] start-line [end-line]]
      [COLUMNS start-column [MAX      ]]
      [                [end-column]]
```

Line and column ranges specified in a SHIFT command only apply to the current command. Specifying lines or columns is optional.

nnn

Number of columns to shift the data.

start-line

Line where the shift begins. You can specify it as:

num Sequence number of the start line.

TOP Indicates the first line of the member. If you do not specify a start line, TOP is the default.

CURSOR Indicates the position where the cursor is located.

offset

A position relative to the top line displayed:

* Top line displayed.

*+**n** Line *n* number of lines below the top line displayed.

*-**n** Line *n* number of lines before the top line displayed.

Note: You can omit the keyword LINE when you specify TOP, BOTTOM, or an offset.

end-line

Line where the shift ends. You can specify it as:

BOTTOM Default. Last line of the member.

num Sequence number.

CURSOR Indicates the position where the cursor is located.

offset Position relative to the top line displayed:

* Top line displayed.

*+*n* Line *n* number of lines below the top line displayed.

*-*n* Line *n* number of lines before the top line displayed.

start-column

Column where the shift begins.

end-column

Column where the shift ends.

MAX

Default. Farthest right-hand column.

Using SET EDIT TRUNCATION Y, you can shift data past the end of the column range (with truncation). Using SET EDIT TRUNCATION N, data that would be truncated is instead positioned at the margin, and the SHIFT proceeds to other lines in the range. This lets you left- or right-align data. This differs from the CHANGE command.

Shift Line Command

The shift line command shifts a line or range of lines by one or more columns to the right or to the left. The shift line command only applies to programs (procedure definition) and data members. You can restrict shift to an inclusive range of lines using the formats below or using the SET EDIT BOUNDS command).

This command has two formats:

()

(())

and

< >

<< >>

The first format () unconditionally shifts data in the column range of 1 to 72. If the shift moves the data out of the edit window, the data is truncated. The second format of the shift line command < > only affects leading or trailing blanks in the column range of 1 to 72.

) or > Shifts a single line to the right.

) or >> Shifts a block of lines to the right. You must enter) or >> on the first and last lines of the block to shift.

)B or >B Lines shifted to the right include this line through the last line.

)T or >T Lines shifted to the right include the first line through this line.

(or < Shift single line to the left.

((or << Shifts a block of lines left. You must enter ((or << on the first and last lines of the block to shift.

(B or <B Lines to shift left include this line through the last line.

(T or <T Lines to shift left include the first line through this line.

Note: You can limit the shift to part of a line or range of lines by placing the cursor on the line being shifted or in the range of lines at the first column you want shifted. Columns to the left of the cursor are not affected.

You can also specify how many columns to shift the data by including a multiplier before or after the command (depending on whether the SET EDIT MULTIPLIER is left or right). With a block shift -- ((,)) , <<, or >> -- the multiplier is specified at the start of the block.

The range of lines to shift does not need to be on the display screen.

Scrolling and SET EDIT TRUNCATION do not affect the margin command.

You cannot use the shift command in sequential dataviews, reports, and data entities.

Templates

This section describes those line commands that display templates of PDL and SQL constructs into the procedure of a program definition. A template contains the key words of a construct with space provided for user supplied information and an indication of what that information is. For example, the following is a template for the IF construct:

```
IF condition
  :statements
ELSE
  :statements
ENDIF
```

The user then supplies the condition and the statements in this construct.

Indentation assists you in following standard indentation practices. Each logical level in the program is indented three columns.

This command has the following format:

```
[n]xx[n]
n
```

Optionally, you can specify the number of statement lines to generate in a template wherever you can insert statements. The SET EDIT MULTIPLIER command determines whether n is placed to the left or right.

For example, specifying 3IF results in the following IF template:

```
IF condition
  :statements
  :statements
  :statements
ELSE
  :statements
  :statements
  :statements
ENDIF
```

xx

Specifies the construct whose template is entered in the procedure. You can specify the following commands:

Line	Command	PDL Construct
	CL	CALL USING
	ER	ERROR PROCEDURE
	FE	FOR EACH
	FF	FOR THE FIRST
	FN	FOR NEW
	IF	IF
	LO	LOOP
	PR	PROCEDURE
	QCC	EXEC SQL CLOSE
	QCM	EXEC SQL COMMIT
	QDC	EXEC SQL DECLARE CURSOR
	QDL	EXEC SQL DELETE
	QES	EXEC SQL SELECT...INTO...
	QFC	EXEC SQL FETCH
	QIN	EXEC SQL INSERT
	QLK	EXEC SQL LOCK TABLE
	QOC	EXEC SQL OPEN
	QRL	EXEC SQL ROLLBACK
	QSQ	EXEC SQL SELECT...GROUP BY...
	QUN	EXEC SQL UNION
	QUP	EXEC SQL UPDATE
	QWH	EXEC SQL WHENEVER
	QXI	EXEC SQL EXECUTE IMMEDIATE
	SA	SET ATTRIBUTE
	SE	SELECT EVERY ACTION
	SF	SELECT FIRST ACTION
	SL	SELECT identifier