

CA Telon® Application Generator

Design Facility Reference Guide

r5.1



This documentation and any related computer software help programs (hereinafter referred to as the "Documentation") are for your informational purposes only and are 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 used or disclosed by you except as may be permitted in 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 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 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 THE END USER 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 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 © 2010 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Product References

This document references the following CA products:

CA Telon® Application Generator (CA Telon)

Contact CA

Contact Technical Support

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

- 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

Provide Feedback

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

If you would like to provide feedback about CA product documentation, complete our short [customer survey](#), which is also available on the CA Support website, found at <http://ca.com/docs>.

Contents

Chapter 1: Introduction	13
Design Facility	13
Generator	14
Using This Guide	16
Audience	16
Where to Find Additional Information	16
Reading Syntax Diagrams	17
 Chapter 2: TDF Main Menu	 25
Accessing the Main menu	25
 Chapter 3: User Profile Maintenance	 29
User Profile Maintenance	29
Update Program Definition Defaults	30
Update Environment Definition Defaults	36
CICS	37
IMS	40
TSO	44
Batch	45
Update Stored Procedure Environment Definition Defaults	47
Update PF Keys Definition	53
Update Session Controls	55
PWS-Specific Features	60
Color Profile	61
PWSCOPY Command	63
PWS Security	63
 Chapter 4: Data Administration	 65
Data Administration Menu	65
List Data Administration Information	70
Create/Update DBD	73
Update DBD Segment	77
List Search Fields	81
Update DLIDSCs For Segment Member	84
Create/Update SSA/Command For DL/I DB/Segment	90
Verify DLIDSC Count	95

Create/Update PSB or File Group	96
Update Sensitive Segment	102
Update Sensitive TLNROWS	104
Show/Update Data Set Default Data	106
Update CICS Queue Default Data	113
Update CICS Journal Default Data	117
Catalog/Import DB2 Tables	120
List SQL Tables/Joins	122
Create/Update SQL Tables/TLNROWS/Temporary Tables	124
Update CA IDMS/SQL CA Datacom/SQL Tables/TLNROWS	132
Update SQL ALIAS	141
Specify Tables Being Joined	143
Update SQL Join—Join Columns	145
Update SQL Join—Access Columns	147
Update SQL Join—Alias	150

Chapter 5: Panel Specification 153

Panel Definition Menu	156
List Panel Definitions	162
Edit Panel Image	165
Line Edit	168
Update Panel Fields (Online)	170
Update Help Message Parameters	176
Update Mapout Parameter	178
Update Attribute Parameter	180
Update Literal Fields	185
Update Output/Input/Outin Field	189
Update Select Fields	199
Update Parameter List Extension	207
List Consistency Edits	209
Update Cross-Field Edit (XFEDIT)	212
Update Segment Edit (SEGEDIT)	215
Consistency Edits Example	222
Create/Update Table SEGLOOP	228
Create/Update File SEGLOOP	233

Chapter 6: Online Program Definition 243

Functionality	243
Online Program Definition Menu	246
Create/Update Screen Definition	250
Create/Update Data Group	267

Update DBMS Characteristics.....	277
Select New Row Name	281
Update Database Segment.....	285
Update Data Set Record	291
Update SQL Detail Data Access	298
Get Diagnostics List.....	309
Get Diagnostics Statement Information	312
Get Diagnostics Condition Information	316
Get Diagnostics Combined Information	320
Fetch Details.....	323
Field Extension.....	327
Update User-Defined Datatypes	328
Select Columns	332
Update DL/I Detail Data Access.....	335
Update Database User I/O SSALIST	341
Update VSAM or Sequential Detail Data Access	343
Update CICS Queue Record	348
Update CICS Journal Record	352
Update CICS Queue Data Access	356
Update CICS Journal Data Access	361
Update TPPCB Detail Data Access	365
List/Show Custom Code	368
Edit Custom Code	371
Update TSO or IMS Screen Environment	373
Update IMS MFSS.....	382
Update PL/I Executable Options	385
Update CICS Screen Environment.....	387
Update/Show Screen Parameters	394
Create/Update Driver Definitions	401
Update IMS/DC Driver Environment.....	410
Create/Update IMS/DC Report Definitions	416
Update IMS/DC Report Environment	423
Update Select Parameters	426
Create/Update Nonterminal Definition	429
List Environments	437
List Stored Procedures To Be Called	439

Chapter 7: Batch Program Definition 441

Update Panel Fields (Batch).....	442
Update Batch Literal Fields.....	446
Update Batch Output Fields	448
Update Panel Group	454

Batch Program Definition	458
Create/Update Batch Definitions	462
Update Batch Environment	475
Update Match Keys	479
List Sorts	481
List Merge Key Groups	483
Update Sort Definition	486
Update Group Merge Keys	493
Create/Update Stored Procedure Definition	494
Update Stored Procedure Environment	501
Update Stored Procedure Parameters	507
Stored Procedure Parameter DBNAME Extension	510

Chapter 8: Prototyping Facility **513**

Prototyping Facility Menu	514
List Panel Definitions	520
List Presentation Stores	523
View Presentation Store	526
View Panel Definition	529

Chapter 9: Utility Functions **533**

Utilities Menu	534
List Headers	540

Chapter 10: Editors and Commands **541**

Panel Editor	541
Invoking the Edit Mode	542
General Rules for Updating Fields	542
Rules for Variable Fields	544
Rules for Literal Fields	544
Rules for Fields with Panel Data	545
Rules for Wrapped Variable Fields	547
Rules for Wrapped Literal Fields	548
Rules for Long Literal Fields	549
Custom Code Editor	549
Presentation Store Editor	551
Primary Commands	552
BACKWARD	553
CANCEL	554
CAPS	554
CH	555

CHANGE	556
COPY.....	558
CREATE	559
EDIT	560
END.....	561
END HOLD	561
EQUATE	562
FIND	562
FORWARD	564
HELP.....	564
HOLD	565
INSTALL	565
ISPF/PDF	565
LEFT	566
LINE EDIT	566
LINE OUT	567
LOCATE	567
MBCOPY (MARK AND BOUND COPY)	568
MBMOVE (MARK AND BOUND MOVE)	569
MBPURGE (MARK AND BOUND PURGE)	570
MENU	570
MOVE	571
NOSHOW	571
NULLS.....	572
PANEL COLUMNS	573
PANEL LINES	573
PANEL SIZE	573
PD	574
PDF	575
PDSCOPY	575
POSITION.....	575
PROFILE	576
PURGE	577
RCHANGE (REPEAT CHANGE)	578
RFIND (REPEAT FIND)	578
REPLACE.....	579
RESET.....	580
RESTORE	580
RESTORE FIND	581
RESUME	582
RIGHT	582
SAVE.....	583

SHOW	584
SHOW — Panel Definition Edit Mode	584
SUBMIT	585
SWAP	585
SWAP EDIT	585
TRANSFER	586
TSO	587
Line Commands	587
CA-Telon Command Summary	598

Appendix A: Field Edit Formats **603**

Standard Field Edit Routines	603
CA Telon Output Edit Routines	603
CA Telon Input Edit Routines	606
Standard Field Edit Modules	609
OCAR	611
OCART	612
OCDATE	612
OCJULIAN	613
ODATE	614
OFLNULL	615
OFLOAT	616
OFORMAT	617
OHEX	617
OHEXA	618
OHEXP	618
OHEXT	619
OINTCDT	620
OINTCJUL	621
OINTLDT	622
OINTLJUL	622
OJULIAN	623
ONULL	624
ONUMNULL	625
OSSA	625
OVCHAR	626
OVNULL	627
IBDOL	628
IBCURRE	629
IBNUM	630
ICDATE	631
ICURRENCY	632

IDATE	633
IDOLLAR.....	634
IFLNULL	635
IFLOAT	636
IFULLCAR	637
IFORMAT	638
IFULLNUM	638
IHEX	639
IHEXA	640
IINTCDT	641
IINTCJUL	642
ICJULIAN	643
IINTLDT	644
IINTLJUL.....	645
IJULIAN	646
ILALPHA	647
ILNULL	648
ILVCHAR.....	648
ILVNULL	649
INBALPHA	650
INBNULL.....	651
INBVCHAR	652
INBVNULL	653
INULL	654
INUMERIC	655
INUMNULL	656
ISTATE	657
IVCHAR.....	658
IVNULL	658
Calls to Field Edit Modules	659
Output edit processing.....	659
Input edit processing	660

Appendix B: Screen Program Identifiers **661**

Screen Program IDs	661
Screen Titles.....	666

Index **673**

Chapter 1: Introduction

CA Telon is a productivity tool that simplifies and speeds the application development process. It reduces the time required for planning, coding, testing, debugging, documenting, and maintaining applications. In this guide, CA Telon Application Generator, formerly known as CA Telon, is referred to simply as CA Telon.

The following CA Telon system components to develop, compile, and test your programs:

- The CA Telon Design Facility (TDF), which helps you to design and code your programs.
- The application system Generator, which translates your TDF-generated CA Telon source statements into a COBOL/LE or PL/I program. See the *Programming Concepts Guide* for complete documentation of the application system Generator.

Design Facility

The CA Telon Design Facility (TDF) is an online tool that you can use to develop individual programs and entire application systems. It produces CA Telon source statements that are input to the CA Telon Generator for creating COBOL/LE, COBOL II, or PL/I source code.

The TDF runs under TSO by means of a CLIST or under CICS as a single transaction. You create programs and application systems by using the TDF to define a panel image, panel definition, and screen definition.

Online help

You can access online help for each screen and the fields it contains. To do this, type **?** in the field for which you need help, and press Enter. The CA Telon Help Facility displays help text.

To leave the Telon Help Facility, invoke END processing.

Panel image

The panel image defines the format of the screen. You create a panel image by keying it on the screen exactly as it should appear to the application user. This process is called *painting* the image.

CA Telon distinguishes among literal fields and fields reserved for input or output by the characters you use to paint the panel image. For example, using installation defaults, a string consisting of the < character represents an input field.

Panel definition

Once you have created a panel image, the next step is to specify the processing characteristics of each field in the panel, such as the source of displayed data, the destination of the entered data, and the field attributes. You can also specify edit criteria, data cross-validation, and key validation. Thus, the panel definition is the field-specific data.

Screen definition

The screen definition consists of general characteristics such as cursor positioning, screen flow, data access, and PF-key processing.

In the screen definition, you can include any custom COBOL/LE, COBOL II, or PL/I code to handle operations specific to your data processing environment or business application. The TDF allows you both to enter custom code and to reference it in the screen definition.

Generator

The CA Telon Generator is a batch job that produces native COBOL/LE, COBOL II, or PL/I source code for application testing or production execution.

Screen definition input

CA Telon source statements developed from the values you input to the TDF are submitted to the Generator. The Generator output is a stand-alone, native COBOL/LE, COBOL II, or PL/I source program. The source program contains all necessary teleprocessing control information.

Generator control

The structure and target environment (for example, CICS, IMS, batch) of the generated program are determined through environment definitions in the TDF.

This architecture allows you to regenerate CA Telon applications simply if performance or environmental considerations change. For example, you can regenerate a CICS DL/I application developed under CA Telon to an IMS/DC application if transaction volume becomes a major concern. Similarly, you can regenerate a conversational IMS/DC application as IMS/DC non-conversational. In addition, during testing you can generate an application as a self-contained unit without BMS control blocks and later generate it with interfaces to BMS along with the BMS source code. The database administrator or technical support can accomplish these changes without any involvement by the application development group.

Program structure

A CA Telon-generated program is easy to follow, partly because it conforms to a standard hierarchical structure.

You can also specify that custom COBOL/LE, COBOL II, or PL/I copybooks are inserted into the generated program exactly as you coded them. This standard structure provides high development and maintenance productivity without sacrificing performance.

The generated program also contains calls to CA Telon-supplied subroutines used for editing and optimizing line traffic. You can also create custom edit subroutines that can be called in generated code.

Since CA Telon generates a native COBOL/LE, COBOL II and above, or PL/I source program, no CA Telon runtime monitor or other "black box" is necessary for program compilation or execution.

See the *Programming Concepts Guide* for details about the structure of the generated program.

Using This Guide

This guide discusses the following information about the CA Telon Design Facility (TDF):

- Each TDF screen, including:
 - Accessing the screen
 - Program ID of the screen
 - Screen function
 - Field description and values for input fields
- TDF editors and editor commands
- Standard CA Telon field edit routines

Audience

This guide is intended for CA Telon application programmers, data administrators, and users who are experienced in data processing concepts and procedures, specifically IMS and CICS concepts, including:

- System administrators and database administrators who support CA Telon
- Application programmers who use CA Telon to develop COBOL/LE, COBOL II and above, and PL/I applications in a mainframe or PWS environment

Use this guide in conjunction with the *Programming Concepts Guide*.

Where to Find Additional Information

You can find additional information about CA Telon in the guides that comprise the CA Telon documentation set. The README file, found in the SAMPJCL PDS on the mainframe and the CA Telon root directory for PWS, provides a list of these guides.

Reading Syntax Diagrams

The formats of all statements and some basic language elements are illustrated using syntax diagrams. Read syntax diagrams from left to right and top to bottom.

The following terminology, symbols, and concepts are used in syntax diagrams.

Keywords

Appear in uppercase letters, for example, COMMAND or PARM. These words must be entered exactly as shown.

Variables

Appear in italicized lowercase letters, for example, *parm1* or *parm2*.

Required Keywords and Variables

Appear on a main line.

Optional Keywords and Variables

Appear below a main line.

Default Keywords and Variables

Appear above a main line.

Double Arrowheads Pointing to the Right

Indicate the beginning of a statement.

Double Arrowheads Pointing to Each Other

Indicate the end of a statement.

Single Arrowheads Pointing to the Right

Indicate a portion of a statement, or that the statement continues in another diagram.

Punctuation Marks or Arithmetic Symbols

If punctuation marks or arithmetic symbols are shown with a keyword or variable, they must be entered as part of the statement or command.

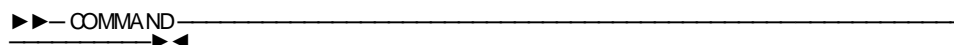
Punctuation marks and arithmetic symbols can include:

,	comma	>	greater than symbol
.	period	<	less than symbol
(open parenthesis	=	equal sign
)	close parenthesis	¬	not sign
+	addition	-	subtraction
*	multiplication	\	division

No Parameters

Below is an example of a statement without parameters.

Statement Without Parameters

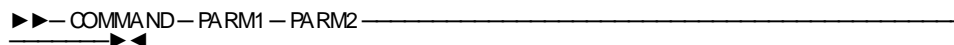


You must write:
COMMAND

Required Parameters

Required parameters appear on the same horizontal line (the main path of the diagram) as the command or statement. The parameters must be separated by one or more blanks.

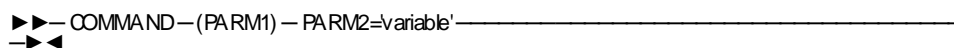
Statement With Required Parameters



You must write:
COMMAND PARM1 PARM2

Delimiters such as parentheses around parameters or clauses must be included.

Delimiters Around Parameters

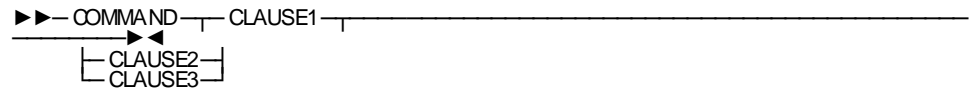


You must write:

COMMAND (PARM1) PARM2='variable'

Where you see a vertical list of parameters as shown below, you must choose one of the parameters. This indicates that only one is required and only one of the displayed parameters is allowed in the statement.

Choice of Required Parameters



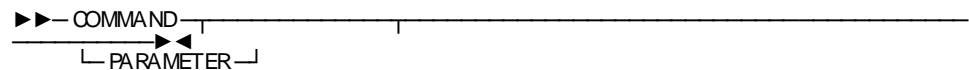
You can choose any of the parameters from the vertical list, such as in the examples below :

COMMAND CLAUSE1
COMMAND CLAUSE2
COMMAND CLAUSE3

Optional Parameters

A single optional parameter appears below the horizontal line that marks the main path.

Optional Parameter



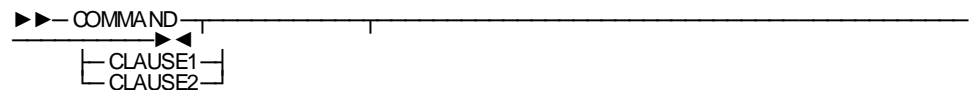
You can choose (or not) to use the optional parameter, as shown in the examples below :

COMMAND

COMMAND PARAMETER

If you have a choice of more than one optional parameter, the parameters appear in a vertical list below the main path.

Choice of Optional Parameters



You can choose any of the parameters from the vertical list, or you can write the statement without an optional parameter, such as in the examples below.

COMMAND

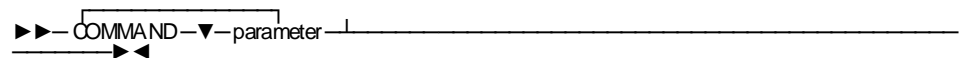
COMMAND CLAUSE1

COMMAND CLAUSE2

Multiple Parameters

For some statements, you can specify more than one parameter, or a single parameter more than once. A repeat symbol (a backward-pointing arrow above the main horizontal line) indicates that you can specify multiple parameters. Below are examples which include the repeat symbol.

Repeatable Variable Parameter



In the above example, the word "parameter" is in lowercase italics, indicating that it is a variable, but it is also on the main path, which means that at least one parameter is required. The repeat symbol indicates that you can specify more than one parameter. Assume that you have three values named PARM-X, PARM-Y, and PARM-Z for the variable. Some of your choices can be:

COMMAND PARM-X

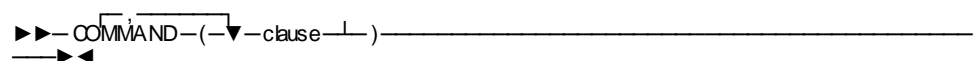
COMMAND PARM-X PARM-Y

COMMAND PARM-Y PARM-Z

COMMAND PARM-X PARM-Y PARM-Z

If the repeat symbol contains punctuation such as a comma, you must separate multiple parameters with the punctuation. Below is an example which includes the repeat symbol, a comma, and parentheses.

Separator with Repeatable Variable and Delimiter



In the above example, the word "clause" is in lowercase italics, indicating that it is a variable, but it is also on the main path, which means that at least one clause is required. The repeat symbol indicates that you can specify more than one clause and that you must separate the clauses with commas. The parentheses indicate that the clauses must be enclosed within parentheses. Assume that you have three values named *CLAUSE-X*, *CLAUSE-Y*, and *CLAUSE-Z* for the variable. Some of your choices can be:

COMMAND (*CLAUSE-X*)

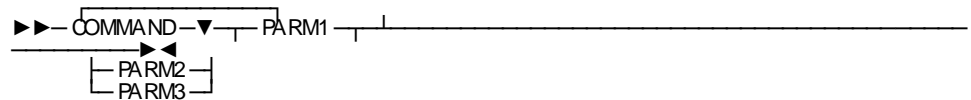
COMMAND (*CLAUSE-X*,*CLAUSE-Y*)

COMMAND (*CLAUSE-Y*,*CLAUSE-Z*)

COMMAND (*CLAUSE-X*,*CLAUSE-Y*,*CLAUSE-Z*)

The following example shows a vertical list of parameters with the repeat symbol.

Choice of Repeatable Parameters



Some choices you can make include:

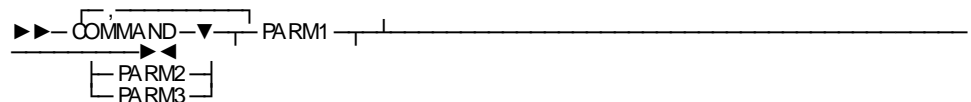
COMMAND PARM2

COMMAND PARM1 PARM2 PARM3

COMMAND PARM3 PARM2 PARM1

The following example shows a vertical list of parameters with a repeat symbol which includes a comma.

Separator with Repeatable Parameters



Some choices you can make include:

COMMAND PARM2

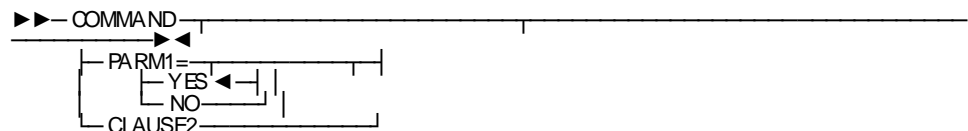
COMMAND PARM1,PARM2,PARM3

COMMAND PARM3,PARM2,PARM1

Default Parameters

When a parameter in a syntax diagram is above the line, for example, YES in the next diagram, its special treatment indicates it is the default parameter. If you do not include the default parameter when you write the statement, the result is the same as if you had actually included the parameter.

Default Parameter



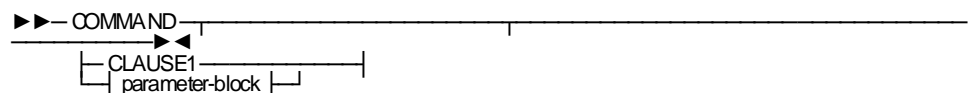
Because YES is the default in the example above, if you write:
COMMAND CLAUSE2

you have written the equivalent of:
COMMAND PARM1=YES CLAUSE2

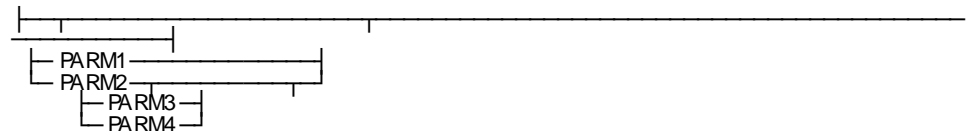
Variables Representing Several Parameters

In some syntax diagrams, a set of several parameters is represented by a single reference, as in the example below.

Variables Representing Several Parameters



Expansion of parameter-block



The *parameter-block* variable can have its own syntax diagram.

Choices you can make from this syntax diagram therefore include (but are not limited to) the following:

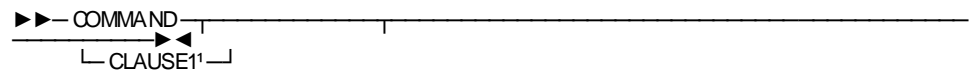
COMMAND CLAUSE1

COMMAND PARM1

COMMAND PARM2 PARM4

Syntax Notes

A note in a syntax diagram is similar to a footnote except that the note appears at the bottom of the diagram box.

Syntax Note

¹ This is a note about the keyword.

Chapter 2: TDF Main Menu

This chapter documents the initial screen of the CA Telon Design Facility, the TDF Main menu. On this menu, you can select the primary design and definition functions for CA Telon:

- User profile maintenance
- Data administration
- Panel definition
- Online program definition
- Batch program definition
- The prototyping facility
- Utilities

Accessing the Main menu

Enter the TSO CLIST, SPF panel, CICS transaction code assigned at installation by your system administrator.

After display of messages related to data set or data file allocation, ending with the display of three asterisks (***), press Enter to display the TDF Main menu.

Program ID

F100

Function

Allows you to select a menu for a CA Telon primary function.

```
TELON DESIGN FACILITY MAIN MENU *****  
COMMAND ==> _____  
  
FUNCTION:  __  
  
          1 — USER PROFILE MAINTENANCE  
          2 — DATA ADMINISTRATION  
          3 — PANEL SPECIFICATION  
          4 — ONLINE PROGRAM DEFINITION  
          5 — BATCH PROGRAM DEFINITION  
          6 — PROTOTYPING FACILITY  
          U — UTILITIES  
          X — EXIT
```

Note: This screen is slightly different for the CA Telon PWS PWS environment. For further information, see the *PWS Option Administrator Guide*.

Field Definitions

COMMAND

For information, see Primary Commands.

Note: The INSTALL command causes a branch to the TDF Installation Menu. A full description of this screen and subordinate screens are found in the "CA Telon Design Facility Customization" section of the *Implementation Guide*.

FUNCTION

The CA Telon primary function selected. CA Telon displays the main menu for the function after you select it. Values are:

1

User profile maintenance. Allows you to define program defaults. Alternately, you can specify a category of defaults here by entering one of these values:

1D Program defaults

1P PF keys

1S Session controls

1C Color Profile (PWS Only)

See User Profile Maintenance for detailed information.

2

Data administration. Allows you to create, update, purge, show, and list PSBs, PCBs in a PSB, databases, segments in a database, VSAM files, queues, journals, or sequential files.

It also allows read-only access to the DB2 catalog. See Data Administration Menu for detailed information.

3

Panel specification. Allows you to create, update, purge, show, and list items related to the panel image and panel definition. See Panel Definition Menu for detailed information.

4

Online program definition. You can create, update, purge, and list items relate to an online screen definition, IMS report definition, driver definition or CICS nonterminal definition. This includes defining the environment, database or data file usage, and custom code. See Online Program Definition Menu for detailed information.

5

Batch program definition. Allows you to create, update, purge, show, and list all items related to a batch program definition.

See Batch Program Definition for detailed information.

6

Prototyping facility. A tool to model either panel images or panel definitions. The application user can view and respond to a prototype of an interactive session with an application function that uses simple scenarios (sequences of screen samples to simulate an application). Prototyping can be performed with or without data mapping. See Prototyping Facility Menu for detailed information.

U

Utilities. You can copy, rename, list, or print any panel image, panel definition, screen definition, report definition, driver definition, nonterminal definition, batch definition, or stored procedure. You can list all headers used in the current TDF, and branch to a list of programs or panels with that header. You can also export your screen, report, driver, nonterminal, batch definition, or stored procedure (the TDF-created definition is translated into a CA Telon source code listing). You can also list all TDF program headers.

See Utilities Menu for detailed information.

X

Exit the TDF and return control to the system.

Chapter 3: User Profile Maintenance

The user profile maintenance option (1) on the TDF Main menu allows you to set default values both for your program and for TDF operations. You can also set or change PF-key values.

This chapter provides information needed to update environment definition defaults, PF keys definition, session controls, and TDF functions specific to PWS.

User Profile Maintenance

Access

On the TDF Main menu, enter **1** in the FUNCTION field.

Program ID

F105

Function

Allows you to select one of the following functions:

- Set default program definition values
- Set TDF PF-key functions
- Set TDF session controls
- Set PWS color profile

USER PROFILE MAINTENANCE MENU *****

COMMAND ==> _____

FUNCTION: —

D — DEFINITION DEFAULTS

P — PFKEYS

S — SESSION CONTROLS

C — COLOR PROFILE

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

FUNCTION

The function that you will perform. Values are:

D

Define default values for input fields on other TDF screens, such as for the transfer work area, custom code name, and the language in which programs are created

P

Define the functions PF keys perform during a TDF session

S

Define default values, such as export parameters and panel image size, that are valid for this and subsequent TDF sessions

C (PWS Only)

Define the colors and intensities for all foreground and background fields on the TDF screens. This option appears only on the PWS. For more information see the *PWS Option Administrator Guide*.

Update Program Definition Defaults

Access

Access this screen in one of the following ways:

- On the User Profile Maintenance menu, enter **D** in the FUNCTION field
- On the TDF Main menu, enter **1D** in the FUNCTION field

Program ID

F112

Function

Sets default values for various input fields on TDF screens.

Note: You can override any default value you supply here on any TDF screen that displays it by simply typing over the default.

When you press Enter or End, your new entries replace the old. You exit the screen by pressing End, which saves all changes and returns you to the previous screen.

```

UPDATE PROGRAM DEFN DEFAULTS *****
COMMAND ==> _____

GENERAL PARAMETERS:
*   LANG      ___      CAPS    ___      APPL    ___
*   OUTIFIL   ___      HELP    ___      HOLD    ___
*   ALARM     ___      EOFKEY   ___

CUSTOM CODE DEFINITION MEMBERS:
*   XFERWKA   _____
*   PGMCUST   _____

EXTENDED ATTRIBUTES:
*   EATTR (Y,N)
*           COLOR    HIGHLIGHT    COLOR    HIGHLIGHT
*   EAIN      ___      EAOUT      ___      ___
*   EALIT     ___      EAERR      ___      ___

UPDATE ENVIRON DEFN DEFAULTS ___ (C-CICS/I-IMS/T-TSO/B-BATCH/S-STORED)

```

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

LANG

The programming language in which CA Telon is to generate programs. This field is not used in installations using only one language. Values are:

COB

COBOL/LE, COBOL II. The default is COBOL/LE.

PLI

PL/I

CAPS

A value to specify whether to translate lowercase characters to uppercase. Values are:

ON

(Default.) CA Telon translates lowercase characters that the application user enters to uppercase.

OFF

No translation occurs on input

APPLID

Contains the optional variable length application id which is used to define the names for programs and control blocks. Its use is determined at installation time. For more information, see the *Implementation Guide*.

OUTIFIL

The initial value for input, outin, and select fields on the screen. Values are:

B

(Default.) Spaces.

U

Underscores

N

Low values

HELP

A value to indicate whether CA Telon is to generate code that you can use to build a HELP facility for the system you develop. Values are:

Y

(Default.) Generate the code.

N

Do not generate the code

HOLD

A value to specify whether CA Telon is to generate code that you can use to build the HOLD capability into the system you develop. Values are:

Y

(Default.) Generate the code.

N

Do not generate the code.

ALARM

A value to specify whether the terminal alarm is to ring automatically when an ERROR-ATTR condition is detected on output. Values are:

Y

Automatic ring on ERROR-ATTR

N

(Default.) No automatic ring on ERROR-ATTR.

EOFKEY (IMS MFS only)

A value to specify whether the application user can erase data by pressing EOF. Values are:

Y

(Default.) Allow the application user to erase data with the EOF key.

N

Do not allow the application user to erase data with the EOF key. MFS does not return to an application program in which a field has been erased using EOF and the attribute for that field has its modified data tag field off.

Therefore, if the EOFKEY value is N, the CA Telon program processes such a field as if it still contained the data originally entered, even if the application user has erased it. If the EOFKEY value is Y for a modified data tag, the MFS for this application is turned on. This results in the resending of every input field to the application (including those that have been erased with EOF).

XFERWKA

The COPY member names to add to the transfer work area section of the program. Each name must be separated by a comma.

PGMCUST

The name of the COBOL section or PL/I procedure in which to add custom code and the name of the custom code added. You can make multiple specifications using this format:

section-name1,member-name1,
section-name2,member-name2,...

Section-name

The four-character identifier of the section or procedure in which to include the custom code (for example, H100) and a suffix (I or T) that specifies whether to include the code at the beginning (I) of the section or procedure, or at the end (T).

For example, H100I specifies section H100 is included at the beginning of the program and E100T specifies section E100 is included at the end of the program.

Member-name

The name of the custom code added at the location specified by *section-name*.

For example, the value A100I, OUTIDC specifies the custom code named OUTIDC is placed at the beginning of the A-100 section.

You can include a member designated in PGMCUSTOM at the beginning and end of any section or procedure in the program, except for U-100 sections or procedures generated by I/O statements (that is, CREATE, DELETE, READ, and UPDATE).

EATTR

A value to specify whether to use extended attributes with this screen. Values are:

Y Use extended attributes

N (*Default.*) Do not use extended attributes. Extended attributes are defined in the EAIN, EALIT, EAOUT, and EAERR fields.

EAIN (color attribute)

The default color attributes for input, output, and select fields. Values are:

- BLUE
- GREEN
- RED
- PINK
- TURQ (Turquoise)
- YELLOW
- NEUTRAL

EAIN (highlight attribute)

The default highlight attributes for input, output, and select fields. Values are:

Value	Meaning
BLINK B BL	Field blinks when displayed.
REVERSE R RE REV REVER REVERS	Field displays in reverse video.

Value	Meaning
DEFAULT	Field appears in default mode.
D	
DE	
DEF	
DEFAU	
DEFLT	
UNDERLINE	Field is underlined.
U	
UN	
UNDER	

EALIT (color attribute)

The default color attributes for literal fields. Values are the same as for the EAIN color attribute.

EALIT (highlight attribute)

The default highlight attributes for literal fields. Values are the same as for the EAIN highlight attribute.

EAOUT (color attribute)

The default color attributes for outin and output fields. Values are the same as for the EAIN color attribute.

EAOUT (highlight attribute)

The default highlight attributes for outin and output fields. Values are the same as for the EAIN highlight attribute.

EAERR (color attribute)

The default color attributes for fields flagged in error. Values are the same as for the EAIN color attribute.

EAERR (highlight attribute)

The default highlight attributes for fields flagged in error. Values are the same as for the EAIN highlight attribute.

UPDATE ENVIRON DEFN DEFAULTS

A value to transfer to the Update Environment Definition Defaults or similar screen for particular target environment. Values are:

C CICS (OS/390)

I IMS

T TSO

B Batch

S Stored

Update Environment Definition Defaults

Access

On the Update Program Definition Defaults screen, enter the character corresponding to the environment updated in the UPDATE ENVIRON DEFN DEFAULTS field.

Program ID

F113

Function

Updates the defaults for the program environment.

UPDATE ENVIRON DEFN DEFAULTS *****				
COMMAND ==> _____				
CICS:	SPASIZE	_____	PSBSCHD	_____
*	TRACE	_____	LINEOPT	_____
*	SPASTG	_____	IOASTG	_____
*	GENPCBS	_____	LNKCOPY	_____
*	DBMS	_____		
				PSBNAME _____
				BMS _____
				TPBSTG _____
				USGCOP1 _____
				USGCOP2 _____
IMS:	SPASIZE	_____	LINKOPT	_____
*	GENPCBS	_____	LNKCOPY	_____
*	TRACE	_____	LINEOPT	_____
*	TRANCODE	_____	TRANMFS	_____
*	WKSPASZ	_____	WKSPAIO: GET	_____
*	WKSPAIN	_____		
				CONVERS _____
				USGCOPY _____
				SPACMPT _____
				TRANFLD _____
				PUT _____
TSO:	GENPCBS	_____	LNKCOPY	_____
BATCH:	TRACE	_____	DLIWGHT	_____
*	GENPCBS	_____	LNKCOPY	_____
*	DBMS	_____		
				USGCOPY _____

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

CICS

SPASIZE

The size of the DFHCOMMAREA used to hold the transfer work area. After CA Telon generates the program, this value appears in the "CICS Program Summary" of the assembler listing as TRANSFER AREA SIZE SPECIFIED. The size must be greater than or equal to the sum of the following items:

- SPA Header
- Application Transfer Work Area
- Screen Image Size

During practical input, estimate this value, and then refine it as needed, based on the requirements printed out on generated program listings.

PSBSCHD

A value to specify whether CA Telon is to automatically schedule and terminate the DL/I PSB in the CICS program sections Q-200-PSB-SCHEDULE and Q-210-PSB-TERM. Values are:

Y

(Default.) Automatically schedules and terminates the DL/I PSB.

N

Generates the Q-200 and Q-210 sections without any reference to them in the CA Telon code. In this case, you are responsible for performing the sections to schedule and terminate the DL/I PSB.

Note: The PSB name used on the schedule call is the variable name (PSB-NAME), which may be dynamically set prior to the PSB scheduling.

PSBNAME

For DL/I, the name of the PSB that the program uses.

TRACE

A value to specify whether the program generates and maintains TRACE variables for debugging. Values are:

Y

(Default.) The program generates TRACE variables.

N

The program does not generate TRACE variables.

Note: TRACE variables increase the size of the generated program.

LINEOPT

The line optimization logic that the program uses and generates. Values are:

1

Use the CA Telon line optimizing subroutine. CA Telon automatically performs line optimization for you.

2

Simulate subroutine optimizing in Custom Code (except for fill character processing).

3

Do not generate line optimizing code.

Note: For CA Telon PWS, LINEOPT must be set equal to 2 or 3.

BMS

A flag to indicate whether the program creates a CICS BMS or uses CA Telon's own mapping. CA Telon's mapping is recommended for all 3270-type terminals. Values are:

Y

Create a BMS map

N

(Default.) Create a CA Telon map.

SPASTG

The location where CA Telon is to generate the SPA-AREA. Values are:

A (AUTO)

- For COBOL, CA Telon generates the SPA-AREA in the Linkage Section. When no SPA-AREA enters a program as it is executed, the program does a GETMAIN to initialize the SPA AREA and the XCTL to the same program.
- For PL/I, when no SPA-AREA enters the program, the program does a GETMAIN, sets the COMPTR, and initializes the SPA-AREA.

S (STATIC)

CA Telon generates the SPA-AREA in COBOL Working Storage.

IOASTG

A COBOL field that specifies where CA Telon generates the SEGMENT-IO-AREA. Values are:

A (AUTO)

Generate SEGMENT-IO-AREA in the Linkage Section and do a GETMAIN for it at program initialization

S (STATIC)

Generate SEGMENT-IO-AREA in Working Storage

TPBSTG

A COBOL field that specifies where CA Telon is to generate the TP-BUFFER. Values are:

A (AUTO)

CA Telon generates the TP-BUFFER in the Linkage Section and performs a GETMAIN for it at program initialization

S (STATIC)

CA Telon generates the TP-BUFFER in Working Storage

GENPCBS

A value to specify whether CA Telon is to include DL/I PCB masks in the program. Values are:

Y

CA Telon automatically generates the PCB masks in the program

N

You must include the PCB masks in the LNKCOPY and

D

The program is a stand-alone program that links USGCOPY TDF fields

LNKCOPY

The name of the COPY member containing the 01-level declarations to include in the Linkage Section.

USGCOP1

A COBOL BLL-POINTER-LIST option specifying the COPY member appended to the BLL-POINTER-LIST entries in the Linkage Section.

DBMS

The name of the default data access for batch programs.

USGCOP2

A COBOL BLL-POINTER-LIST option specifying the COPY member included in the Q-100-CICS-INIT to initialize the BLL-POINTER-LIST entries.

IMS

SPASIZE

The size of the SPA specified in the IMS generation for this application. It can be defined in conjunction with WKSPASZ. The size of SPASIZE alone or of SPASIZE plus WKSPASZ must be large enough to hold:

- The SPA header
- The application transfer work area
- The largest screen image in the application

Find the initial value for this field based on SPA requirements printed on the generated program listings.

LINKOPT

The type of linking to occur in the IMS program. Values are:

D

Indicates the program is a stand-alone program that links dynamically to other programs. If you do not specify ANY for LINKPGM or MSGPGM, all programs to which control can be passed must be accounted for in the MSGTRAN, MSGPGM, MSGTBL, or LINKPGM fields. See Online Program Definition Menu for more information.

Any attempt to transfer control to an unspecified module results in a user abend.

S

CA Telon generates the program as a subroutine that executes under a driver. The only valid fields with this option are CONVERS, MFSMOD, and PGMNAME. See Online Program Definition Menu for details about these fields.

CONVERS

A value indicating whether the system that CA Telon generates is IMS conversational. Values are:

Y

Specifies that the system being generated is an IMS conversational system using an IMS SPA

N

Specifies that the system being generated is an IMS non-conversational system using a WORKSPA database

GENPCBS

A value to specify whether to include DL/I PCB masks in the program. Values are:

Y

Automatically generate PCB masks in the program

N

PCB masks must be included in the LNKCOPY and USGCOPY members

LNKCOPY

The name of the COPY member containing the 01-level declarations to include in the Linkage Section.

USGCOPY

The variable declarations in the linkage member appended to the COBOL Procedure Division or the PL/I procedure statements.

In COBOL, the variable declaration is the list of 01-level variables in the Linkage Section. In PL/I, the variable declaration is a list of DECLARE statements in the Linkage Section.

TRACE

A value to specify whether the program generates and maintains trace variables for debugging. Values are:

Y

(Default.) The program generates TRACE variables.

N

The program does not generate TRACE variables.

Note: Trace variables increase the size of the generated program.

LINEOPT

The line optimization logic that the program uses and generates. Values are:

1

Use the CA Telon line optimizing subroutine. CA Telon automatically performs line optimization for you.

2

Simulate subroutine optimizing in custom code (except for fill character processing).

3

Do not generate line optimizing code.

Note: For CA Telon PWS, LINEOPT must be set equal to 2 or 3.

SPACMPT

A value to indicate whether CA Telon should generate a SPA with a fixed number of overhead bytes, making it compatible for use by both static and dynamic programs. The generation of such a SPA allows for message switching between static and dynamic modules in an application. Values are:

Y

CA Telon generates a compatible SPA. CA Telon generates a field called SPA-CMPAT in the dynamic programs where the next-program-name field exists in static programs.

N

CA Telon generates SPA with a different number of overhead bytes, based on whether the module is static or dynamic. Message switching between static and dynamic modules cannot take place in the application.

TRANCDE

The name of the IMS transaction code associated with the generated program, specified only when the IMS transaction is different from the CA Telon-generated program name. The CA Telon-generated program is *hhnnnn*, where:

- *hh* is the header
- *nnnn* is the ID as entered on the program definition

TRANMFS

A value to specify whether the MFS processes the transaction code. Values are:

Y

MFS processes the transaction code

N

(*Default.*) MFS does not process the transaction code.

TRANFLD

The transaction code that is imbedded to the MID for this screen, specified only when the /FORMAT command is used to start the application, or if the system is non-conversational. See the *Programming Concepts Guide* for additional information.

WKSPASZ

The number of bytes in the WORKSPA database used in this application system, specified only when you use a WORKSPA database.

In a non-conversational system, the value must be large enough to hold the application transfer work area, overhead, and the largest screen image in the application.

In a conversational system, the value must be large enough to hold the remainder of this area overflowing from the IMS SPA area (defined with the value in the SPASIZE field).

GET

The name of the COPY member containing custom code included before the read of a WORKSPA database.

PUT

The name of the COPY member containing custom code included before the replacement of a WORKSPA database.

WKSPAIN

A value to specify whether the generation of CA Telon WORKSPA database initialization code is to take place in the IMS program section C-920-GET-WORKSPA. Values are:

Y

(Default.) Results in generation of code to reinitialize the CA Telon transfer work area when the IMS program flow has been broken.

N

Results in no such generation of code

TSO

GENPCBS

A value to specify whether to include DL/I PCB masks in the program that CA Telon generates. Values are:

Y

Automatically generate PCB masks in the program

N

Include PCB masks in the LNKCOPY and USGCOPY members

LNKCOPY

The name of the COPY member containing the 01-level declarations to include in the Linkage Section.

USGCOPY

The variable declarations in the linkage member appended to the COBOL Procedure Division or the PL/I procedure statements.

In COBOL, the variable declaration is the list of 01-level variables in the Linkage Section. In PL/I, variable declaration is a list of DECLARE statements in the Linkage Section.

Batch**TRACE**

A value to specify whether the program generates and maintains trace variables for debugging. Values are:

Y

(Default.) The program generates TRACE variables.

N

The program does not generate TRACE variables.

Note: Trace variables increase the size of the generated program.

DLIWGHT

A value to specify whether the CA Telon program generates automatic weighting of DL/I calls. Values are:

Y

Generate automatic weighting of DL/I calls

N

Do not generate automatic weighting of DL/I calls

CA Telon increments a generated variable by one (1) for each DL/I REPL call and by three (3) for each DL/I DLET or ISRT call. The programmer is responsible for checking and resetting the variable to determine when a checkpoint should be taken.

GENPCBS

A value to specify whether to include DL/I PCB masks in the program that CA Telon generates. Values are:

Y

Automatically generate PCB masks in the program

N

Include PCB masks in the LNKCOPY and USGCOPY members

LNKCOPY

The name of the COPY member containing the 01-level declarations to include in the Linkage Section.

USGCOPY

The variable declarations in the linkage member appended to the COBOL Procedure Division or the PL/I procedure statements.

In COBOL, the variable declaration is the list of 01-level variables in the Linkage Section. In PL/I, variable declaration is a list of DECLARE statements in the Linkage Section.

DBMS

The name of the default data access for batch programs. Values are:

- DL/I
- VSAM
- SEQ
- DB2
- IDMS SQL
- DATACOM

Update Stored Procedure Environment Definition Defaults

Access

To access the Update Stored Procedure Environment Definition Defaults screen, on the Update Program Definition Defaults screen, enter an **R** in the UPDATE ENVIRON DEFN DEFAULTS Field.

Program ID

F154

Function

Specifies the default environment data for the Stored Procedure programs.

```

UPDATE STOR ED ENVIRON DEFN DEFAULTS *****
COMMAND ==> _____

RESULTS _____ ASUTIME _____ WLMENV _____
SCHEMA _____ COLLID _____

PARAMETERS:
* EXTSCUR: __ (2-DB2 U-USER D-DEFINER)
* PRIMSTYL: __ (N-GENERAL WITH NULLS G-GENERAL D-DB2SQL J-JAVA)
* PROGTYP: __ (M-MAIN S-SUB)
* SQLMOD : __ (M-MODIFIES SQL DATA N-NO SQL S-CONTAINS SQL
               R-READS SQL DATA)

SELECT ALL THAT APPLY (Y/N):
__ COMRETN      __ DBINFO      __ DETERM
__ FENCED      __ NULCALL      __ STAYRES
  
```

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

RESULTS

Specifies the maximum number of result sets that the stored procedure can return. The default is 0, which indicates that there are no result sets.

ASUTIME

Specifies the total amount of processor time, in CPU service units, that a single invocation of a stored procedure can run.

WLMENV

Identifies the MVS Workload Manager (WLM) environment in which the stored procedure is to run when the DB2 stored procedure address space is WLM-established. The name of the WLM environment is a long identifier.

Note: If a WLM ENVIRONMENT is not specified, the stored procedure runs in the default WLM-established stored procedure address space specified at installation time.

SCHEMA

Part of the qualified name for the stored procedure.

The qualified form of stored procedure name is a short SQL identifier (the SCHEMA name) followed by a period and a long SQL identifier.

COLLID

Identifies the package collection to be used when the stored procedure is executed. This is the package collection into which the DBRM that is associated with the stored procedure is bound.

If COLLID is not specified, the package collection for the stored procedure is the same as the package collection of the calling program.

EXTSCUR

Specifies how the stored procedure interacts with an external security product, such as RACF, to control access to non-SQL resources.

2—The stored procedure does not require a special external security environment. This is the only valid choice when a Workload Manager environment is not specified (DB2).

U— An external security environment should be established for the stored procedure. Access is performed using the authorization ID of the user who invoked the stored procedure (USER).

D— An external security environment should be established for the stored procedure. Access is performed using the authorization ID of the stored procedure owner (DEFINER).

PRMSTYL

Identifies the linkage convention (parameter style) used to pass parameters to the stored procedure. All of the linkage conventions provide arguments to the stored procedure that contain the parameters specified on the CALL statement. Some of the linkage conventions pass additional arguments to the stored procedure that provides more information to the stored procedure.

D

(DB2SQL) In addition to the parameters on the CALL statement, the following arguments are also passed to the stored procedure:

- Null indicator for each parameter on the CALL statement
- SQLSTATE to be returned to DB2
- Qualified name of the stored procedure
- Specific name of the stored procedure
- SQL diagnostic string to be returned to DB2
- DB2INFO structure, if DBINFO is specified

G

(GENERAL) Only the parameters on the CALL statement are passed to the stored procedure. The parameters cannot be null.

N

In addition to the parameters on the CALL statement, another argument is passed to the stored procedure. The additional argument contains a vector of null indicators for each of the parameters on the CALL statement that enables the stored procedure to accept or return null parameter values (GENERAL WITH NULLS).

J

The stored procedure uses a convention for passing parameters that conforms to the Java and SQLJ specifications. IN OUT and OUT parameters are passed as single-entry arrays. The DBINFO structure is not passed (JAVA).

PROGTYP

Specifies whether the stored procedure runs as a main routine or a subroutine.

S

The stored procedure will run as a subroutine (SUB).

M

The stored procedure will run as a main routine (MAIN).

SQLMOD

Indicates whether the stored procedure can execute SQL statements and, if so, what type it can execute.

N

The stored procedure cannot execute any SQL statements (NO SQL).

M

The stored procedure can execute any SQL statement except those statements that are not supported in any stored procedure (MODIFIES SQL DATA).

R

The stored procedure cannot execute SQL statements that modify data. SQL statements that are not supported in any stored procedure return a different error (READS SQL DATA).

S

The stored procedure cannot execute any SQL statements that read or modify data. SQL statements that are not supported in any stored procedure return a different error (CONTAINS SQL).

COMRETN

Indicates whether DB2 commits the transaction immediately on return from the stored procedure (COMMIT ON RETURN).

N

DB2 does not issue a COMMIT when the stored procedure returns.

Y

DB2 issues a COMMIT when the stored procedure returns if the following statements are true:

- SQLCODE that is returned by the CALL statement is not negative.
- Stored procedure is not in a must-abort state.
- COMMIT operation includes work that is performed by the calling application process and the stored procedure.

If the stored procedure returns result sets, the cursors that are associated with the result sets must have been defined as WITH HOLD to be usable after the COMMIT.

DBINFO

Specifies whether specific information known by DB2 is passed to the stored procedure when it is invoked.

N

Additional information is not passed.

Y

An additional argument is passed to the stored procedure when it is invoked. This argument is a structure that contains information such as the application run-time authorization ID, the schema name, the name of a table or column that the procedure might be inserting into or updating, and identification of the database server that invoked the procedure.

DETERM

Specifies whether the stored procedure returns the same result from successive calls with identical input arguments.

Y

The stored procedure returns the same result from successive calls with identical input arguments (DETERMINISTIC).

N

The stored procedure might not return the same result from successive calls with identical input arguments (NOT DETERMINISTIC).

FENCED

Specifies whether the stored procedure runs in an external address space (to prevent user programs from corrupting DB2 storage).

Y

The stored procedure will run in an external address space (FENCED).

N

The stored procedure will not run in an external address space (NOT FENCED).

NULCALL

Determines whether the stored procedure is called even when any of the input arguments are null, making the procedure responsible for testing for null argument values.

Y

The stored procedure is called even when any of the input arguments are null; the stored procedure is responsible for testing for null argument values.

N

The stored procedure is not called when any of the input arguments are null.

STAYRES

Specifies whether the stored procedure load module remains resident in memory when the stored procedure ends.

Y

The load module remains resident in memory after the stored procedure ends.

N

The load module is deleted from memory after the stored procedure ends.

Update PF Keys Definition

Access

Access this screen in one of the following ways:

- On the User Profile Maintenance menu, enter **P** in the FUNCTION field
- On the TDF Main menu, enter **1P** in the FUNCTION field

Program ID

F110

Function

Defines the functions that the PF keys perform while you are using the TDF.

When you press Enter or EOF, your new entries replace the old. You exit the screen by pressing End, which saves all changes and returns you to the previous screen.

```

UPDATE PRKEYS *****
COMMAND ==> _____

PF/PA KEY DEFINITIONS:
PF1 ==> _____ PF13 ==> _____ PA1 ==> _____
PF2 ==> _____ PF14 ==> _____ PA2 ==> _____
PF3 ==> _____ PF15 ==> _____ PA3 ==> _____
PF4 ==> _____ PF16 ==> _____ CLEAR ==> _____
PF5 ==> _____ PF17 ==> _____
PF6 ==> _____ PF18 ==> _____
PF7 ==> _____ PF19 ==> _____
PF8 ==> _____ PF20 ==> _____
PF9 ==> _____ PF21 ==> _____
PF10 ==> _____ PF22 ==> _____
PF11 ==> _____ PF23 ==> _____
PF12 ==> _____ PF24 ==> _____

VALUES:  HOLD    SWAP    END HOLD  HELP    FORWARD  BACKWARD
          END     MENU    LINE EDIT CANCEL  SAVE      NOP
          LEFT   RIGHT  LINE OUT  LOCATE  RFIND     RCHANGE
          CAPS   NULLS  PROFILE  RESET   RESTORE   SWAP EDIT
          RESUME PDF    ISPF    CAN     PD        END ALL
LINE COMMANDS: D I R C M A B O ) ( X XX F L COLS FS LC G U DG
  
```

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

PF/PA KEY DEFINITIONS

A set of commands associated with PF keys.

Values are presented on the screen as VALUES and LINE COMMANDS. Default values are displayed in the designated fields.

See Primary Commands for information about VALUES. See Line Commands for information about line commands.

The value NOP means "no option." When you press a PF key which has been assigned the NOP value, no function is performed.

You can change the PF-key functions at any time. Any changes you make are in effect once you exit the screen. They remain in effect until you change them again.

Note: You cannot define the PA keys and Clear in TSO. If you do, they will not function.

DEFAULT HEADER

The one- to five-character header used with the names of all programs and control blocks generated for a particular application. Defining it here causes it to appear on every TDF screen that has a HEADER field.

The default is a two-digit header specified at installation.

IMAGE CHARACTERS

The characters that CA Telon uses to identify the usage of a field that you paint on a panel image. Values are any character except a single quote (') and an ampersand (&). However, each character used to define a field type may not be used in a literal field on a panel image, except for the literal break character. Default values are:

<	Input field
>	Output field
+	Outin field
	Select field
\	Literal break

The usage of each of these types of fields is:

Literal

A literal field is any character or series of characters whose display does not vary during program execution. Data comprising a literal field is mapped out to the screen but is not otherwise used in processing.

Input

An input field is any character or series of characters keyed by the application user and processed as input from the screen. In most cases, the field is mapped to a database or work area field. Each input field can be mapped to two separate data fields.

Output

An output field is any character or series of characters that the program displays to the application user. The data for an output field originates from a database, data file, or work field, and is protected from modification by the application user.

Outin

An outin field is a combination output and input field that is processed during program output. The application user can change it, then it is processed during the input phase.

Select

A select field is a special input field that allows the application user to designate the next screen for processing. When the application user enters data into a select field, the program branches to a screen or process predetermined by the programmer. When a screen has more than one select field, the application user must enter data in only one select field. The program displays an error message when data is entered into more than one field.

Literal Break

The literal break character breaks a literal field into two parts. Use this character when you want to process parts of a literal or control an attribute.

FIELD SELECTION

The fields that are displayed on the Edit Panel Image screen when you update a panel image. Values are:

Y

Display fields of this type

N

Do not display fields of this type

Default values are:

Y

Input

Y

Output

Y

Outin

Y

Select

N

Literal

Do not modify the default values unless required for special applications.

COMPRESS STATEMENTS

A value to specify whether the TDF is to generate compressed CA Telon source statements. Values are:

Y

Compress statements

N

(*Default.*) Generate one statement per line.

ENVIRONMENT

The execution environment for the program. Values are:

B

BATCH

C

CICS (OS/390)

I

IMS

T

(*Default.*) TSO.

FORMAT

The environment format under which the program runs at the Generator request of the CA Telon program. Values are:

Y

Environment format request wanted

N

(*Default.*) No environment format request.

See Utilities Menu for detailed information about this field.

PSB

The environment PSB (CICS, IMS, or TSO) the program runs under at the Generator request statement section of the CA Telon program. Values are:

Y

Environment PSB requested

N

(*Default.*) Environment PSB not requested.

See Utilities Menu for detailed information about this field.

DITTO CHARACTER

A special character that enables you to repeat the previous statement on many TDF screens. Values are any character *except* an asterisk (*). The default is the double quote (").

If you enter the ditto character in a field and then press Enter, CA Telon repeats the field immediately above.

PANEL SIZE

The number of lines and columns on a screen, in the format *//ccc* where:

//

The number of lines on the screen or report (maximum = 99). The default value is 24.

ccc

The number of columns on a screen or report (maximum = 240) The default value is 80.

The total screen size must not exceed 9920. To calculate screen size, multiply the total number of lines by the total number of columns on the screen.

Note: If you are working with a Model 3 or Model 4 terminal, CA Telon displays the panel in a Model 2 mode if you create a panel with 24 lines.

UPPER CASE LITERALS

A value to specify whether the literals can contain lowercase characters. Values are:

ON

(Default.) All literals converted to uppercase.

OFF

Lower case literals remain as keyed

USER MODE

A value to specify whether the program definition menus are presented in short form or long form. Values are:

1

Short form

2

Long form

See the following sections in this manual for more information:

- Online Program Definition Menu
- Batch Program Definition

DG SEPARATORS

The character used in the first and second separator lines of the Update Data Access screen. Values include any character. If you do not specify a value, hyphen (-) is the default.

Note: If you specify '¬', no separator is displayed.

PDSCOPY DSNAME (Mainframe only)

The default data set searched when you enter the PDSCOPY command with only a member name. See Primary Commands for information about the PDSCOPY command.

PWS-Specific Features

These TDF functions are specific to PWS. They are provided in addition to the standard mainframe TDF features and functions.

- Color Profile - The color profile function enables you to select, as part of your TDF profile, the colors and intensities for all foreground and background fields on the TDF screens.
- PWSCOPY Command - A command in the Custom Code Editor that allows you to copy ASCII files from a directory into the custom code member. PWSCOPY is equivalent to PDSCOPY on the mainframe.
- Security - The PWS provides a security layer that controls access to its menus and dialogs.
- Mouse - You may use your mouse to position the cursor on input fields in the TDF.

Note: The ISPF, TSO, and PDF commands are not available in the PWS.

Color Profile

On the TDF Main menu, enter **1C** in the FUNCTION field.

Program ID

F116

Function

Allows you to select colors for the TDF screens. This program is only available for PWS.

COLOR PROFILE *****
COMMAND ==> _____

ENTER COLORS BELOW:

		FOREGROUND (0-15)	BACKGROUND (0-7)	BLINKING (Y/N)	RESTORE DEFAULTS (Y)
		=====	=====	=====	=====
PROTECTED NORMAL	==>	--	--	--	--
PROTECTED HIGH	==>	--	--	--	--
UNPROTECTED NORMAL	==>	--	--	--	--
UNPROTECTED HIGH	==>	--	--	--	--
STATUS LINE	==>	--	--	--	--
SCREEN COLOR	==>	--	--	--	--
.					
.					
.					

Field Definitions

COMMAND

See Primary Commands for information.

Protected Normal

All literal text that appears on the screen.

Protected High

Usually the error message field.

Unprotected Normal

Fields where input can be entered.

Unprotected High

Fields where input has been entered in error.

Status Line

Short help message and related fields at bottom of screen.

Screen Color

Blank areas of the screen.

Foreground

The color that displays as the foreground for the field type. Values are 0 through 15.

Background

The color that displays as the background for the field type. Values are 0 through 7.

Blinking

Specifies if the field should blink or not. Values are:

Y

The field should blink

N

The field should not blink

Restore defaults

Returns the field type definition to the default and overrides the current color combinations.

Y

Return the field type to the default definition

N

Leave the field type definition

PWSCOPY Command

This command enables you to copy an ASCII text file into a TDF custom code editor member. The command format is:

```
PWSCOPY file-name  
or  
PWSCOPY *
```

Values are:

file-name

The drive, directory, and fully qualified name of the file

*

Allows you to select the file using the File Open Dialog box. After selecting the file, click OK and the file is copied into the custom code member at the location specified by the A (After) or B (Before) line command.

PWS Security

PWS includes security to administer the CA Telon work environment from the TDF. This allows administrators to control access TDF functions and to various PWS menu items for three user types:

- Controller
- Programmer
- Analyst

For more information about PWS security, see the appropriate chapter in the *PWS Option Administrator Guide*.

Chapter 4: Data Administration

Using data administration, Option 2 on the TDF Main menu, you can define and maintain the following CA Telon entities:

- File group
- Program specification block (PSB)
- Database description (DBD)
- SQL table
- SQL join
- VSAM file
- Sequential file
- CICS queue
- CICS journal

This chapter provides technical reference information for the data administrator. See the *Data Administration Guide* for additional information about using the data administration option.

Data Administration Menu

Access

On the TDF Main menu, enter **2**.

Program ID

D100

Function

Specifies an administration function (definition or maintenance data) to be performed on a named CA Telon entity.

This menu allows you to CREATE, UPDATE, PURGE, SHOW, LIST, or CATALOG:

- A file group (FG)
- A Program Specification Block (PSB)
- DL/I Database Description (DL)
- An SQL table (TB)
- An SQL join (TJ)
- A VSAM file (VS)
- A sequential file (SQ)
- A CICS queue (CQ)
- A CICS journal (CJ)

DATA ADMINISTRATION MENU *****					
COMMAND ==> _____					
FUNCTION:	---	CR-CREATE	UP-UPDATE	PU-PURGE	SH-SHOW
		CA-CATLG/DB2			LI-LIST
ITEM:	---	FG-FILE GRP	PS-PSB		
		DL-DLI DBD	TB-SQL TBL	TJ-SQL JOIN	
		VS-VSAM	SQ-SEQ FILE	CQ-CICS QUE	CJ-CICS JRNL
NAME:	_____	(QUAL.TBLNAME/TLNNAME FOR SQL ITEMS)			
DESC:	_____				
BASE:	_____	(QUAL.TBLNAME/TLNNAME FOR SQL ITEMS) (USED ONLY FOR FUNCTION: CR)			

Considerations

- Not all FUNCTIONS apply to all ITEMS.
- You must define a DBD on the Create/Update DBD screen before you can use it in a PSB.
- You can define a file group (FG) in the same way as a PSB, except that you can also reference data sets. PSBs and file groups are defined on the Create/Update PSB Or File Group screen.
- You can import PSBs or DBDs from source definitions by using the utilities described in Utilities Guide or in the *PWS Option Administrator Guide*.
- The CA Telon data administration function does not interface directly to a data dictionary. Therefore, you must import information from DBD and PSB source statements, or directly from the catalog for table definitions.

To import DBD and its PSB source statements, use the import jobs described in Utilities Guide or in the command files described in the *PWS Option Administrator Guide*.

- To import CA-IDMS/SQL and CA-Datcom/SQL table definitions, use their respective SQL Extract utilities and the Transport_In utility.
- Use the CA Telon data administration function to provide default information to simplify the programmer's task of creating program definitions.

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

FUNCTION

The function that you will perform. Values are:

CR

Create the entity specified in the ITEM field. For this function, NAME and DESC are required fields.

UP

Create the entity specified in the ITEM field. For this function, NAME is a required field.

PU

Purge the entity specified in the ITEM field. For this function, NAME is a required field.

SH

Show the entity specified in the ITEM field. For this function, NAME is a required field.

LI

List the entity specified in the ITEM field. The list of entries begins with the mask value, if any, specified in the NAME field.

CA

List tables in the DB2 catalog to select and import tables from the catalog into the TDF.

ITEM

The type of entity you are defining or maintaining. Values are:

FG

File group

PS

Program specification block (PSB)

DL

Database description (DBD)

TB

SQL table or view

TJ

SQL join

VS

VSAM file

SQ

Sequential data set

CQ

CICS queue

CJ

CICS journal

NAME

The name of the entity you are defining or maintaining. The name must be one to eight characters long. However, if you specify the SQL name (*qualifier.name*) instead of the CA Telon name, *qualifier* can be one to eight characters long and *name* can be one to 18 characters long.

Note: The length of CICS transient data (TYPE=TD) queue names is limited to four characters.

DESC

A description of the entity being defined or maintained. This field is for informational purposes only but is required when the FUNCTION field value is CR.

BASE

The name of an existing entity whose definition you are modifying to create a new definition. This value is ignored when the FUNCTION field value is *not* CR.

For SQL, you can specify the *CA Telon-name* or *qualifier.table.name* of an existing table or join.

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

** (FUNCTION) A

The function to perform on an item. Enter one of the values in the next table in the first column, to the left of the NAME field for the entity.

Value	Function	Description
C	Copy	Copies a member
D	Description	Changes the description of a member
P	Purge	Purges a member and confirms the purge
R	Rename	Renames a member
S	Show	Allows you to browse a member
U	Update	Allows you to enter a member in update mode
Z	Zap	Purges a member without confirming the purge

NAME

The name of the entity.

RENAME

An additional function being performed on the entity. A value is required if the value in the FUNCTION field is C or R.

If you rename an entity, you must access it by its new name.

Values are listed in the next table with the confirmation message, if any, that is displayed in the RENAME field after the function is performed.

Value	Description	Message
C	Copies a member	*COPIED
D	Changes the description of a member	*DESC UP
P	Purges a member and confirms the purge	
R	Renames a member	*RENAMED
S	Allows you to view the member in browse mode	
U	Allows you to enter a member in update mode	*PROCSD
Z	Purges a member without confirming the purge	*PURGED

DESCRIPTION

The description of the entity, used for informational purposes only. The value in this field is saved only if the value in the FUNCTION field is D.

USER

The user or terminal ID that most recently modified the entity. CA Telon automatically maintains the value in this field.

UPDATE

The date of the last update of the entity. CA Telon automatically maintains the value in this field.

Create/Update DBD

Access

On the Data Administration menu, enter:

- **CR** or **UP** in the FUNCTION field
- **DL** in the ITEM field
- Name of the DBD in the NAME field
- Description of the DBD in the DESC field

Program ID

D111

Function

Updates IMS database segments.

XXXXXXXXX CREATE DBD *****
COMMAND ==> _____ PAGE 01

ACCESS

RMNAME

SEQ	TYPE	NAME	PARENT/DEVICE	MAX LTH	SEGMENT KEY	LENGTH	START
---	-----	-----	-----	-----	-----	-----	-----
---	-----	-----	-----	-----	-----	-----	-----
---	-----	-----	-----	-----	-----	-----	-----
---	-----	-----	-----	-----	-----	-----	-----
---	-----	-----	-----	-----	-----	-----	-----
---	-----	-----	-----	-----	-----	-----	-----
---	-----	-----	-----	-----	-----	-----	-----
---	-----	-----	-----	-----	-----	-----	-----
---	-----	-----	-----	-----	-----	-----	-----
---	-----	-----	-----	-----	-----	-----	-----
---	-----	-----	-----	-----	-----	-----	-----
---	-----	-----	-----	-----	-----	-----	-----
---	-----	-----	-----	-----	-----	-----	-----
---	-----	-----	-----	-----	-----	-----	-----
---	-----	-----	-----	-----	-----	-----	-----
---	-----	-----	-----	-----	-----	-----	-----
---	-----	-----	-----	-----	-----	-----	-----
---	-----	-----	-----	-----	-----	-----	-----

Accessing the Show/Purge Screen

You can access the Show/Purge DBDs (D117) screen from the Data Administration menu by entering:

- SH or PU in the FUNCTION field
- DL in the ITEM field
- Name in the NAME field

Alternately, on the List Data Administration Information screen (D401) you can enter S or P as a line command for the desired table.

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

ACCESS

(Protected field.) Identifies the access mode that the application uses to read the databases (for example, HISAM or HDAM). A value is displayed only if the DBD was imported.

RMNAME

(Protected field.) Specifies information that CA Telon uses to manage data in a fast path DBD or in the root segment of an HDAM database. A value is displayed only if the DBD was imported into the TDF.

SEQ

A sequence number for the segment. You can also enter one of these line commands in this field:

C

Copy a line

CC

Copy a block of lines

I

Insert a line

Inn

Insert *nn* lines

M

Move a line

MM

Move a block of lines

R

Repeat a line

RR

Repeat a block of lines

A

Line(s) to insert, copy, or move go after this line

B

Line(s) to insert, copy, or move go before this line

D

Delete an entry

U

Transfer to the Update DBD Segment screen

See Primary Commands for more information.

TYPE

The statement types used by CA Telon to generate a DBD. Values are:

DATA SET

The DATA SET= statement, used to define a data set group in a database. The value in this field is taken in conjunction with the NAME and DEVICE field values.

LCHILD

The LCHILD= statement, used to define logical relationships in a logical database. You must define this field in a physical database or databases.

SEGM

The SEGM= statement, used to define:

- The segment type (with the NAME field value)
- The position of the segment in the hierarchy of the database (with the PARENT field value)
- The length of the data portion of the segment type (with the MAX LTH field value)

At least one SEGM= statement must follow a DATA SET= statement. The SEGM field defines the DBD statement.

NAME

The value in this field is:

- The primary data set in the data set group, when the value in the TYPE field is DATA SET. The value in this field defines the DD1= value in the DBD DATA SET statement.

Note: Duplicate names are not allowed in a DBD generation for IMS.

- The segment name, when the TYPE field is SEGM. The value, taken in conjunction with the PARENT field value, defines the NAME= value in the DBD SEGM statement.

PARENT/ DEVICE

The value in this field depends on the value in the TYPE field:

- For EXEC DLI processing, the name of the parent segment for this segment, field when the value in the TYPE field is SEGM. If this is a root segment, the PARENT value must be zero or blank. Otherwise, this field must contain the name of the parent segment. CA Telon uses the value in this field to generate the correct DL/I call and to generate the PSB. It defines PARENT=, a DBD SEGM statement.
- The device on which you want the data set group stored, when the TYPE field is DATA SET. The value in this field defines DEVICE=, a DBD DATA SET statement.

MAX LTH

For DL/I path processing, a value to define BYTES=, a DBD SEGM statement, when the TYPE field is SEGM.

SEGMENT KEY

The key field (IMSKEY) for the segment. This value defines the first subfield of NAME=, a DBD FIELD statement.

Note: For a non-keyed segment, leave this field blank.

LENGTH

The number of bytes that the key in the segment occupies. This value defines BYTES=, a DBD FIELD statement.

START

The starting byte position of the key field in the segment. This value defines START=, a DBD FIELD statement.

Update DBD Segment

Access

On the Create/Update DBD screen, enter **U** in the SEQ field next to a DBD segment.

Program ID

D112

Function

Specifies default information for a segment.

UPDATE DBD: _____

SEGM: _____

COMMAND ==> _____

OPTIONS ==> SEARCH FIELDS PCB PARMS DLIDSC

GENERAL: LABEL _____

* COPY _____ COPYLV1 COPYLBL : _____

A

** DLIDSC SEGMENT CMND IMSKEY OP KEY

Field Definitions

UPDATE DBD

(Protected field.) Displays the DBD name specified in the NAME field on the Data Administration menu.

SEGM

(Protected field.) Displays the SEGM name specified in the NAME field of the Create/Update DBD screen.

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

SEARCH FIELDS

Search fields defined for this segment, if the displayed value is a plus sign (+).

To proceed, enter any non-blank value. This transfers you to the List Search Fields screen.

PCB PARMS

The PCB parameters. To select this field, enter a non-blank value. This transfers you to the Extended Parameter Utility screen, on which you can enter parameters such as PROCOPT and INDICES. See :hhref refid=d215. for information on PROCOPT and INDICES.

DLIDSC

The DLI data search criterion. To proceed, enter any non-blank value. This transfers you to the Update DLIDSCs For Segment Member screen.

LABEL

The default label for the segment name. CA Telon uses this field when generating the name of the COBOL paragraph or PL/I procedure that contains the DL/I call. If you do not specify a value, the default is the name of the segment referenced in the DBD.

COPY

The COBOL COPY member name or PL/I %INCLUDE member name that contains the segment definition.

If you enter a member name, CA Telon uses the contents of that member for the layout of the segment, as in this example:

```
DBSEG    TRGEMPL  
USAGE    INREAD  
COPYLBL  TRG1-IO-AREA  
COPY     TRG1IO
```

If you enter **NONE**, CA Telon does not copy a segment definition into the program, as in this example:

```
DBSEG    TRGEMPL  
USAGE    INREAD  
COPYLBL  TRG1-IO-AREA  
COPY     NONE
```

Note: NONE is not valid for index data sets.

If you do not specify a value, the default is the value of the SEGM field, as in this example:

```
DBSEG    TRGEMPL  
USAGE    INREAD  
COPYLBL  TRG1-IO-AREA  
COPY     (uses segment name)
```

For VSAM processing, the member name is the same as the data set name, on the Create/Update PSB Or File Group screen or the Show/Update Data Set Default Data screen.

When you use the COPY and COPYLBL fields, if there is more than one segment of the same name (under different PCBs), you must specify a unique COPY/COPYLBL combination, as in this example:

```
DBSEG    TRGEMPL
USAGE    INREAD
COPYLBL   TRG2-IO-AREA
COPY      (uses segment name)
```

COPYLV1

A value to specify that the COPY/INCLUDE member for the I/O area of this segment is to start at the COBOL or PL/I 01 level. Values are:

Y

The segment is to start at the COBOL or PL/I 01 level. You must also specify the COPYLBL field to supply the I/O area for the data access that CA Telon generates.

N

CA Telon generates the 01 and 02 levels and uses the COPY/INCLUDE member for 03 levels and below.

COPYLBL

The name of the COPY or INCLUDE member containing the segment definition. This value overrides the CA Telon default *IOA-segment-name-SEGMENT*.

If the member has the same name as the segment (see the SEGM field), you may omit this value and CA Telon will include the member whose name matches the segment name.

If you enter **NONE**, no segment definition is copied into the program.

If you have specified Y in the COPYLV1 field and CA Telon is generating automatic I/O, a COPYLBL value is required.

If two segments have the same segment name (the default value of DBSEG or another name that you specify in the LABEL field) and the same COPYLBL, CA Telon generates only one I/O area. Thus, for CA Telon to generate a second I/O area, specify a unique data item name in the COPYLBL field.

(Sequence number) A

(*Protected field.*) Displays sequence numbers.

DLIDSC

(*Protected field.*) Identifies the DLI data search criterion (DLIDSC) at this segment level, used to generate the SSA.

The CA Telon-reserved names for DLIDSCs are:

- NONE
- DFLT
- QUAL
- UNQUAL
- **DFLT**
- **QUAL**
- **UNQUAL**

SEGMENT

(*Protected field.*) Displays the value specified in the NAME field on the Create/Update DBD screen.

CMND

(*Protected field.*) Displays *---.

IMSKEY

(*Protected field.*) Displays the value obtained from the SEGMENT KEY field on the Create/Update DBD screen. This value defines NAME=, a DBD FIELD statement.

OP

(*Protected field.*) Displays =. However, if the IMSKEY field is blank, the OP field is also blank.

KEY

The PL/I or COBOL variable name containing the key to this segment.

If inheritance is requested at the key level for this segment in the program definition, this value is retrieved and suffixed to the inheritance sign (@) in the program definition. For example, if the value in this field is WK-KEY, the *inherited key* in the program definition is @WK-KEY. See the *Programming Concepts Guide* for more information on inheritance. See also the description of update field shown next.

Note: If the IMSKEY field is blank, this field must also be blank.

(Update field)

A value to specify that you are updating the key value. Enter any non-blank value, except for a plus sign (+), to transfer to a general-purpose screen on which you can update the KEY value at the lowest segment level (the last line displayed on this screen).

A plus sign (+) in this field indicates that the value in the KEY field is more than 30 bytes. In this case, the KEY field is protected, and you must enter a non-blank character in the update field to update the KEY value.

List Search Fields

Access

Access this screen in one of the following ways:

- On the Update DBD Segment screen, enter any non-blank character in the SEARCH FIELDS field
- On the Create/Update SSA/Command For DL/I DB/Segment screen, enter any non-blank character in the U field

Program ID

D115

Function

Maintains as many as 160 IMS database segment search fields.

XXXXXXXX LIST SEARCH FIELDS ***** *****
COMMAND ==> _____ Page 01

SEGMENT	NAME	_____	PARENT	_____	MAX LTH	_____
*	IMSKEY	_____	LENGTH	_____	START	_____
*	KEYPIC	_____			TYPE	_____

A SRCHFLD NAME LENGTH START TYPE KEYPIC

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

NAME

(*Protected field.*) Identifies the name of the segment defined on the Create/Update DBD screen.

PARENT

The name of the parent segment for this segment, as specified in the PARENT field on the Create/Update DBD screen.

If this is a root segment, the PARENT value must be zero. Otherwise, the PARENT must be the name of a segment previously defined for the same DBD. CA Telon uses this field for setting up the correct DL/I call and for generating the PSB.

MAX LTH

For DL/I path processing, a value to define BYTES=, a DBD SEGM statement. This value was specified in the MAX LTH field on the Create/Update DBD screen.

IMSKEY

The key field for the segment. This value defines the first subfield of NAME=, a DBD FIELD statement. This value was specified in the SEGMENT KEY field on the Create/Update DBD screen.

Note: For a non-keyed segment, leave this field blank.

LENGTH

The number of bytes that the key in the segment occupies. This value defines BYTES=, a DBD FIELD statement. This value was specified in the LENGTH field on the Create/Update DBD screen.

START

The starting byte position of the key field in the segment. This value defines START=, a DBD FIELD statement. This value was specified in the START field on the Create/Update DBD screen.

KEYPIC

The format of the key for the segment. CA Telon uses this value to define the VALUE field in the SSA.

If the format of the key for the segment is a character format, do not enter a value.

If the format of the key for the segment is packed decimal or binary and the key value to be moved to the SSA is of a different format, the key is converted when it is moved to the SSA.

The following examples are COBOL entries in the KEYPIC field:

9(5) COMP-3

X(5)

S9(4) COMP

9(8)V9(7)

TYPE

The TYPE of data that the search field contains. Values are:

X

Hexadecimal

P

Packed

C

Character, alphanumeric, or numeric

F

Binary full-word

H

Binary half-word

(Sequence number)A

The sequence number of the field. You can also use this field as a select field.

The field is protected until you enter a value in the SRCHFLD NAME field.

SRCHFLD NAME

The name of the field used in the segment search argument (SSA) for DL/I applications.

LENGTH

The length of the search field that is used when the SSA is generated.

START

The starting position of the search field in the segment.

TYPE

The type of data that the search field contains. Values are as documented for the first TYPE field on this screen.

KEYPIC

The format of the key for the segment, as documented for the first KEYPIC field on this screen.

Update DLIDSCs For Segment Member

Access

On the Update DBD Segment screen, enter any non-blank character in the DLIDSC field.

Program ID

D116

Function

Lists segment search argument (SSA) information that you can modify.

```
UPDATE DLIDSCS FOR SEGMENT MEMBER *** *****
COMMAND ==> _____ SCROLL ==> ____
DBD: _____ SEGMENT: _____

***** DLIDSC  USEONT  CMND  IMSKEY  OP      KEY          MORE
A  _____
```

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

DBD

(*Protected field.*) Identifies the name of the DBD that you are listing.

SEGMENT

(*Protected field.*) Identifies the name of the segment that you are listing.

(Line command) A

The first column of each line, allowing entry of a line command to modify the associated item(s). Values are:

C

Copy a line.

I

Insert a line

R

Repeat a line.

A

Line(s) to insert, copy, or move go after this line.

B

Line(s) to insert, copy, or move go before this line.

D

Delete an entry.

S

Select a DLIDSC for a segment level, or select a user I/O request from updating the program definition data group. This command is available only when control has been transferred to this screen from the Update Database Segment screen or the Update DL/I Detail Data Access screen. This command increments the use count by 1; use the V line command to check the use count.

U

Select the line to update an SSA (control is transferred to the Create/Update SSA/Command For DL/I DB/Segment screen).

V

Verify the program definitions that reference the DLIDSC (control is transferred to the Verify DLIDSC Count screen and the use count, specified in the USECNT field, is updated).

Note: Use of this command prevents all other users from updating a program definition data group until the verify process has completed.

See Line Commands for more information.

DLIDSC

The DLI data search criterion (DLIDSC) at this segment level, used to generate the SSA for DL/I. To access this field, enter **I** as a line command.

The first character must be a letter, @, #, or \$. You can specify numbers as subsequent characters. No other special characters are allowed and if specified, will produce a Generator error.

The value in this field may not match the segment name.

The CA Telon-reserved names for DLIDSCs are:

- NONE
- DFLT
- QUAL
- UNQUAL
- **DFLT**
- **QUAL**
- **UNQUAL**

To create a DLIDSC that accesses a secondary index:

1. Enter the **I** line command.
2. Enter the DLIDSC field value.
3. Update the line.

PROCSEQ and the remaining SSA information are entered on the Create/Update SSA/Command For DL/I DB/Segment screen.

USECNT

(*Protected field.*) Specifies one of the following:

- The number of times a data group specified in the program definition references the DLIDSC
- If the value is 1, a program definition

CMND

The SSA command code field for IMS, overriding the default on the segment CA Telon generates.

The character you enter is prefixed with an asterisk (*) and suffixed with one to three hyphens (-) to make a 4-byte code. For example, if you enter **D** in this field, CA Telon generates the SSA command code parameter as:

*D- -

If you do not specify a value, CA Telon uses the IMS default *---.

For EXEC DLI, you can use this field as a shorthand method of specifying qualification options. However, note that EXEC DLI does not support all command codes.

IMSKEY

The name of the key used in qualifying the SSA for the segment. This value defines the first subfield of NAME=, a DBD FIELD statement.

Note: For an unqualified SSA, leave this field blank.

You can update the value entered here on the Create/Update SSA/Command For DL/I DB/Segment screen.

OP

For DL/I processing, the relational operator for the SSA that CA Telon generates.

If not specified, the default is **>=** for BROWSE data access and **=** for keyed access.

The values for DL/I processing are as follows:

Value	Description
=	Equal to
>=	Greater than or equal to
=>	Greater than or equal to
<=	Less than or equal to
=>	Less than or equal to
>	Greater than

Value	Description
<	Less than
^=	Not equal to
=^	Not equal to

Note: The OP field value does not apply to unqualified SSAs; that is, for BROWSE requests for which there is no starting key value (STBRKEY, defined on the Create/Update File Segloop screen). If you use the SCHFLDx fields on the Create/Update File Segloop screen, OP may be (but need not always be) set to =.

OPCODE Override Rules

The following rules apply to OPCODE overrides:

- User I/O OPCODE overrides the segment-level OPCODE
- Segment-level OPCODE overrides the DLIDSC OPCODE in data administration

CA Telon generates the value of the *segment-name*-SSA-OPCODE in Working Storage based on the segment-level OPCODE override.

If you override the OPCODE at the user I/O before the DL/I call in U-100, CA Telon moves the overridden OPCODE to xxxx-SSA-OPCODE and resets it to the OPCODE in Working Storage after the DL/I call.

If the user I/O OPCODE is the same as the segment-level OPCODE, CA Telon does not generate MOVE statements for the OPCODE.

KEY

The name of the host variable value used for the data access.

Note: The inheritance symbol (@) is not valid because inheritance is not valid at this level.

MORE

(*Protected field.*) Indicates whether there is more information than is shown on this line. Possible values are:

+

There is more information than is shown on this line

X

Information about secondary index processing for the DLIDSC

Enter **U** as a line command on this screen to transfer control to the Create/Update SSA/Command For DL/I DB/Segment screen and to see the additional information.

Create/Update SSA/Command For DL/I DB/Segment

Access

On the Update DLIDSCs For Segment Member screen, enter **U** as a line command.

Note: This access is unavailable if you are using only the default DLIDSC. Therefore, on the Update DLIDSCs For Segment Member screen, first enter **I** as a line command to insert another DLIDSC and update it.

Program ID

D118

Function

Creates or updates your SSA definitions, allowing you to specify default SSA/command information for a segment in a DL/I database. CA Telon uses this information to create DLIDSC statements at export time.

UPDATE SSA/CMD FOR DL/I SEGMENT * *****									
COMMAND ==> _____									
DBD: _____	SEGMENT: _____	DLIDSC: _____	PROCSEQ: _____						
GENERAL: KEYFEED _____									
*	CMDCODE _____	-OR-	PATH (Y/N) _____	CURRENT (Y/N) _____	OPTION (F/L) _____				
*			CONCATK (Y/N) _____	PARENTG (Y/N) _____	LOCKED (Y/N) _____				
EX DL/I: VARLTH (Y/N) _____	OFFSET _____								
WHERE/BOOLEAN SSA: _____									
U	IMSKEY	OP	KEY					BOOL	OP
_____	_____	_____	_____	_____	_____	_____	_____	_____	_____

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

DBD

(*Protected field.*) Identifies the name of the DBD that you are listing.

SEGMENT

(*Protected field.*) Identifies the segment that you are listing.

DLIDSC

(*Protected field.*) Identifies the DLI data search criterion (DLIDSC) at this segment level that CA Telon uses to generate the SSA for DL/I or qualification options for EXECDLI.

Note: An asterisk (*) is not allowed as a first character in a DLIDSClabel.

PROCSEQ

The secondary index that is processed for this segment through program definition. PROCSEQ only allows the *parents* of LCHILDS in the DBD.

KEYFEED

The name of the host variable in which to place the concatenated key that EXECDLI returns. This field is ignored for DL/I.

CMDCODE

The SSA command code field for IMS, overriding the default on the segment CA Telon generates.

The character you enter will have a four-byte code consisting of:

- A prefix of an asterisk (*)
- A suffix of three hyphens (---)

For example, if you enter **D** in this field, CA Telon generates the SSA command code parameter as:

*D--

If you do not specify a value, CA Telon uses the IMS default *---.

For EXEC DLI, you can use this field as a shorthand method of specifying qualification options. However, note that EXEC DLI does not support all command codes.

PATH

A value to specify whether CA Telon is to retrieve the segment referred to by this DLIDSC as part of a path call. Values are:

Y

This SSA is used in standard path call generation (generates the D command code)

N

This SSA is used in path call processing (generates the N command code)

Blank

This SSA is a standard IMS SSA

CURRENT

A value to specify whether CA Telon is to maintain positions for all levels above the segment type. Values are:

Y

Generate the command code

N

Do not generate the command code

OPTION

The command option FIRST or LAST, equivalent to a DL/I call function command code. Use these command options only with GET and ISRT type commands. Values are:

FIRST

Equivalent to the DL/I F call function command code. Use it to specify the first occurrence of the segment type that satisfies the rest of the qualification in the command.

LAST

Equivalent to the DL/I L call function command code. Use it to specify the last occurrence of the segment type that satisfies the rest of the qualification in the command.

CONCATK

A value to specify whether to use qualification statements or a concatenated key to qualify an I/O request. This value is equivalent to specifying or not specifying the C command code. Values are:

Y

Generate the C command code

N

Do not generate the C command code

PARENTG

The current PARENT level. This is equivalent to specifying the EXEC DLI SETPARENT option and the command code. Values are:

Y

Generate the command code

N

Do not generate the command code

LOCKED

A value to specify whether the segment is locked by the program, even if the program does not update the segment. This prevents a segment from being altered by other programs until the program reaches the next sync point and field is equivalent to the command code. Values are:

Y

Generate the command code

N

Do not generate the command code

VARLTH

A value to specify whether the segment is variable length.

OFFSET

The offset to the parent segment in the I/O area. Specify the offset or the relative byte length from the beginning of the record. For example, if the key starts on the fourth byte, enter **3**. You can specify this field only when using logical relationships.

U

A value to transfer control to the List Search Fields screen so that you can select a search field (if, for example, you cannot remember the name of the search field). You can select a search field and it is updated to this screen.

Enter **U** or **S**.

IMSKEY

The name of the key used in qualifying the SSA for the segment. This value defines the first subfield of NAME=, a DBD FIELD statement.

Note: For an unqualified SSA, leave this field blank.

OP

For DL/I processing, the relational operator for the SSA that CA Telon generates.

If not specified, the default is >= for BROWSE data access and = for keyed access.

For DL/I processing, values are:

Value	Description
=	Equal to
>=	Greater than or equal to
=>	Greater than or equal to

Value	Description
<=	Less than or equal to
=>	Less than or equal to
>	Greater than
<	Less than
^=	Not equal to
=^	Not equal to

The OP field value does not apply to unqualified SSAs; that is, for BROWSE requests for which there is no starting key value (STBRKEY, defined on the Create/Update File Segloop screen). If you use the SCHFLDx fields on the Create/Update File Segloop screen, OP may be (but need not always be) set to =.

KEY

The name of the host variable name whose value is moved to the SSA for the database call.

BOOLOP

The Boolean operator to connect qualification statements. Values are:

AND

Dependent AND (changed to **&** for DL/I processing)

*

Dependent AND (changed to **AND** for EXEC DL/I processing)

OR

Logical OR (changed to **+** for DL/I processing)

+

Logical OR (changed to **OR** for EXEC DL/I processing)

|

Logical OR (changed to **OR** for EXEC DL/I processing)

#

Independent AND (not valid for EXEC DL/I processing)

Verify DLIDSC Count

Access

On the Update DLIDSCs For Segment Member screen, enter **V** as a line command.

Program ID

D120

Function

Displays the program definitions that use a particular SSA/command.

```

XXXXXXXX  VERIFY DLIDSC COUNT *****
COMMAND ==>
DBD: _____ SEGMENT: _____ DLIDSC: _____

USED BY      USED BY      USED BY      USED BY      USED BY      USED BY
_____
  
```

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

DBD

(*Protected field.*) Identifies the name of the DBD that you are listing.

SEGMENT

(*Protected field.*) Identifies the segment that you are listing.

DLIDSC

(*Protected field.*) Identifies the DLI data search criterion (DLIDSC) at this segment level that CA Telon uses to generate the SSA for DL/I. This value was specified on the Update DLIDSCs For Segment Member screen.

USED BY

(*Protected field.*) Displays the program definitions that use a particular SSA/command.

Create/Update PSB or File Group

Access

On the Data Administration menu screen, enter:

- **CR** or **UP** in the FUNCTION field
- **FG** or **PS** in the ITEM field
- *Name* in the NAME field
- *Description* in the DESCRIPTION field

Program ID

D211

Function

Defines or updates file groups or IMS program specification blocks (PSBs).

[illegible]

Show/Purge screen

You can access the Show/Purge PSBs Or File Groups screen from the Data Administration menu screen by entering:

- **SH** or **PU** in the FUNCTION field
- **FG** or **PS** in the ITEM field

- *Name* in the NAME field
- *Description* in the DESCRIPTION field

Alternatively, on the List Data Administration Information screen, you can enter **S** or **P** as a line command for a PSB (/P) or file group (/G).

Field Definitions

The Show/Purge PSBs Or File Groups screen fields are the same as the Create/Update PSB Or File Group screen fields.

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

LANG

(Informational purposes only.) The default language option. Values are:

- COBOL
- COB
- PLI
- PL/I

COMPAT

A value to specify whether the I/O PCB is included when the PSB is added to the data group with DGADD from the Create/Update Data Group screen. Values are:

Yes

The PSB is treated as if it were an I/O PCB.

No

(Default.) The PSB has an I/O PCB added when in a BMP or MSG region.

If the DBMS field is equal to EXEC DLI, you must specify GENPCBS=N. This means that you are responsible for defining all of the PCBs, including the I/O PCB. You should define the I/O PCB as IO-PCB (or IO_PCB for PLI) to avoid compile errors.

SEQ

An edit field. Editor commands are:

C

Copy a line

CC

Copy a block of lines

I

Insert a line

II

Insert *nn* lines

M

Move a line

MM

Move a block of lines

R

Repeat a line

RR

Repeat a block of lines

A

Line(s) to insert, copy, or move go after this line

B

Line(s) to insert, copy, or move go before this line

D

Delete an entry

U

Select the line to update a database

If you enter **U**, CA Telon transfers control to:

- The Update Sensitive Segment screen, if TYPE is DB
- The Update Sensitive TLNROWS screen, if TYPE is SQL or TABLE

See Line Commands for more information.

TYPE

The type of program control block (PCB) or other item being specified in the PSB or file group. When specifying items for a PSB, values are:

TP or TPPCB

IMS teleprocessing PCB. Generates a CA Telon TPPCB statement.
Required for all alternate PCBs.

GSAM

IMS database access. Required for all GSAM database PCBs.

DB or Database

IMS and DL/I database access. Required for all DL/I database PCBs.

When specifying a file group, values are:

TP or TPPCB

IMS teleprocessing PCB. Generates a CA Telon TPPCB statement.
Required for all alternate PCBs.

GSAM

IMS database access. Required for all GSAM database PCBs.

DB or Database

IMS and DL/I database access. Required for all DL/I database PCBs.

DS or Data Set

CICS or batch VSAM file access, or batch sequential file access.

SQL

SQL tables.

TABLE

SQL tables.

CQ

CICS temporary storage or transient data queues.

QUEUE

CICS temporary storage or transient data queues.

CJ

CICS journals.

JOURNAL

CICS journals.

NAME

The name of the item in the PSB or file group. The next tables show how to determine the name:

Item	Name
PCB	DBDNAME associated with this PCB
VSAM or sequential file	DDNAME associated with the file

Item	Name
SQL table	TLNNAME associated with the table
CICS queue or journal	Name assigned when the queue or journal is defined to the TDF

Option	PCB Type	Description
Name	Alternate	The destination of the message or transaction code
	Database	The DBD used as the primary source of the database segments
	GSAM	The GSAM database PCB
Type	Alternate	Logical terminal name

PCBNAME

The CA Telon COBOL name used to access this PCB.

CA Telon generates the PCB name by suffixing the characters that you specify in this field with -PCB. If you do not enter a value in this field, CA Telon generates the name *dbd-name-PCB*, where *dbd-name* is the value in the NAME field.

For example, if the NAME value is GCPVM2T and no value is specified in the PCBNAME field, the PCB name in the generated programs is GCPVM2T-PCB. However, if the value in the PCBNAME field is PROVIDER, the PCB name is PROVIDER-PCB.

KEYLEN

The number of bytes in the PCB key feedback area. The value in this field is specified in KEYLEN=, a PCB statement.

Note: When no information is available in the data group, the default value for KEYLEN is 100.

PROCSEQ

The name of the secondary index used to process the database named in the PCB statement's DBDNAME operand. The value in this field is specified in PROCSEQ=, a PCB statement.

PROCOPT

The DB processing options. This field is valid only when the value in the TYPE field is DB or GSAM. The value in this field is specified in PROCOPT=, a PCB statement.

For more information about the DB processing options, see the documentation of IMS utilities.

EXP

The setting of the EXPRESS value in the PCB that CA Telon generates, to indicate whether the PCB processes a message if an application abends. This field is valid only for teleprocessing PCBs. Values are:

Y

Messages from the alternate PCB are sent if an ABEND occurs

N

Messages from the alternate PCB are backed out if an ABEND occurs

The default value depends upon the value of the ABC field: if the ABC value is **Y**, the EXP value defaults to Y; otherwise, the default value is N.

ABC

A setting to indicate that the PCB that CA Telon generates replaces CA Telon's XFER-PCB in calls to the CA Telon ABEND handling routine. This field is valid only for teleprocessing PCBs. Values are:

Y

(*Default.*) The PCB that CA Telon generates replaces CA Telon's XFER-PCB

N

CA Telon uses the XFER-PCB

PRT (PRINT)

A value to specify whether the REPORT subroutine uses this PCB. If you do not code this field, then the print subroutine uses the XFER-PCB. This field is valid only for teleprocessing PCBs. Values are:

Y

The PCB that CA Telon generates replaces CA Telon's XFER-PCB.

N

(*Default.*) CA Telon uses the XFER-PCB.

LTERM

The destination for messages sent using this PCB. The value in this field is specified in LTERM=, a PCB statement.

Values include any IMS transaction code and any logical terminal name defined to IMS. If not specified, the destination for the PCB is modifiable in the program.

Update Sensitive Segment

Access

On the Create/Update PSB Or File Group screen, enter **U** in the SEQ field for databases.

Program ID

D215

Function

Updates values used to generate the SENSEG statement.

The PSB authorizes which sensitive statements you can view .

XXXXXX UPDATE SENSITIVE SEG ***** *****
COMMAND ==> _____ PAGE 01

PCBNAME			DBNAME	
SEQ	NAME	LABEL	PROCOPT	INDICES
---	----	-----	-----	-----
---	----	-----	-----	-----
---	----	-----	-----	-----
---	----	-----	-----	-----
---	----	-----	-----	-----
---	----	-----	-----	-----
---	----	-----	-----	-----
---	----	-----	-----	-----
---	----	-----	-----	-----
---	----	-----	-----	-----
---	----	-----	-----	-----
---	----	-----	-----	-----
---	----	-----	-----	-----
---	----	-----	-----	-----
---	----	-----	-----	-----
---	----	-----	-----	-----
---	----	-----	-----	-----
---	----	-----	-----	-----
---	----	-----	-----	-----
---	----	-----	-----	-----

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

You can also enter the **INIT** command to reinitialize all segment characteristics for the database. Even if only one segment is displayed on this screen, INIT reinitializes all of the segments on the database.

PCBNAME

(Protected field.) Specifies the CA Telon COBOL name used to access this PCB. This value was defined on the Create/Update PSB Or File Group screen in the PCBNAME field or the NAME field.

DBNAME

(Protected field.) Specifies the field keyword for various types of PCBs. This value was defined on the Create/Update PSB Or File Group screen in the NAME field.

SEQ

A sequential number for the sensitive segment. You can also enter the **D** line command to delete the line.

NAME

The name of the sensitive segment. This value is specified in NAME=, a SENSEG statement.

LABEL

The default label for the segment name. CA Telon uses this field when generating the name of the COBOL paragraph or PL/I procedure that contains the DL/I call. If you do not specify a value, the default is the name of the segment referenced in the DBD.

PROCOPT

The PROCOPT value to include in the corresponding SENSEG statement for the PSB that CA Telon generates.

If you do not specify a value, CA Telon generates the PCB PROCOPT field based on data access. For example, an OUTREAD data access causes CA Telon to generate a PROCOPT of G. The value in this field does not affect the program that CA Telon generates.

INDICES

The value to include in the corresponding SENSEG INDICES field that CA Telon generates for the PSB. If you do not specify a value, CA Telon does not generate a value for the INDICES field.

Update Sensitive TLNROWs

Access

In the Create/Update PSB Or File Group screen, enter **U** in the SEQ field for a table.

Program ID

D216

Function

Deletes sensitive TLNROWs, causing CA Telon to delete the TLNROW from the table definition.

```
XXXXXX  UPDATE SENSITIVE TLNROWS ** *****  
COMMAND ==> _____ PAGE 01  
FGNAME _____ TBLNAME _____  
  
DELETE OR MOVE TLNROWS TO CREATE PROPER SENSITIVITY  
SEQ TLNROW  C-CNT      COLUMNS DEFINED  
_____  
_____
```

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

You can also enter **INIT** to reinitialize all TLNROW characteristics for the database. Even if only one TLNROW is displayed on this screen, INIT reinitializes all of the TLNROWs on the database.

FGNAME

(*Protected field.*) Displays the file group name specified in the NAME field on the Data Administration menu.

TBLNAME

(*Protected field.*) Contains the TLNNAME for the table.

SEQ

A field in which you enter **D** to delete an entry from the table. CA Telon eliminates the TLNROW from the table definition.

TLNROW

(*Protected field.*) Identifies the CA Telon defined rows in your table. For each row, there are potentially multiple columns.

Note: The first row must have the name in the LABEL field on the Create/Update SQL Tables/TLNROWs screen.

C-CNT

(*Protected field.*) Specifies the number of columns in the row.

COLUMNS DEFINED

(*Protected field.*) Identifies the columns defined for this TLNROW.

If a plus sign (+) is displayed to the right of this field, there is more column information for this TLNROW. To view the information, access the Create/Update SQL Tables/TLNROWs screen.

Show/Update Data Set Default Data

Access

Access this screen in one of these ways:

- On the List Data Administration Information screen, enter **S** or **U** as a line command
- On the Data Administration menu, enter:
 - **SH** or **UP** in the FUNCTION field
 - **VS** or **SQ** in the ITEM field
 - *Name* in the NAME field
 - *Description* in the DESCRIPTION field

Program ID

D114

Function

Specifies data set record default information for a file group that includes this data set for a program definition CA Telon uses to initialize the respective data set information in a data group.

```

XXXXXXXX  UPDATE DATA SET RECORD DEFAULTS *** *****
COMMAND ==> _____ PAGE 01
DATA SET _____ ACCESS _____

GENERAL:  LRECL  _____ (MIN MAX)  BLKSIZE _____ COMMIT _____
*         OPEN  _____ (INPUT/OUTPUT/I - O/EXTEND/UPDATE)

RECORD:   LABEL  _____
*         COPY   _____
*         COPYLV1 (Y/N) _____
*         COPYLBL _____
*         COBDIV  (FD/WS) _____
*         COBVSky _____

I/O:
*         KEY    _____
*         KEYLEN _____
*         OPCODE _____

VSAM:     TYPE   _____ (KSDS/RRDS/ESDS)  ACCMODE _____ (DYN/RAN/SEQ/DIR)
*         OPTLIST _____
*         RECLTH _____
*         GENKEYL _____ INDEXOF _____ REUSE (Y/N)
  
```

Field Definitions

Note: Fields 14 through 23 are displayed only if you enter **VS** (VSAM) in the ITEM field.

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

DATA SET

(*Protected field.*) Identifies the data set to which you are specifying default information.

ACCESS

The type of access for the data set. Values are VSAM and SEQ (sequential).

LRECL

The length (in bytes) of the logical record for the data set. For variable-length files, enter the minimum length in the first portion of the field (MIN) and the maximum length in the second portion of the field (MAX). For fixed length files, either leave MAX blank or enter the same value as you do in MIN.

BLKSIZE

The blocking factor (in bytes) of the data set. If you do not specify a value, you must specify the blocking factor in the JCL.

COMMIT

See the U\$TNAS4. for more information.

OPEN

The way in which CA Telon automatically opens the data set at program initialization. Values are:

- INPUT
- OUTPUT
- I/O (VSAM and sequential only)
- EXTEND
- UPDATE

At program termination, CA Telon closes any files automatically opened at program initialization.

LABEL

A replacement for the name of this data set in the host variable names that CA Telon generates.

COPY

The COBOL COPY or PL/I INCLUDE member name that contains the record layout. Enter either a member name or **NONE** in this field.

CA Telon uses the contents of the member for the layout of the records. If you enter **NONE**, CA Telon does not copy a record layout member into the program.

If the COPY or INCLUDE member has the same name as the record, you need not specify a value in this field.

Note: The COPY field is not valid for indexes.

COPYLV1

A value to specify whether or not the COPY/INCLUDE member for the I/O area of this record starts at the COBOL or PL/I 01 level. Values are:

Y

The COPY or INCLUDE member is included in its entirety at the COBOL or PL/I 01 level. You must also specify the COPYLBL field to supply the I/O area for the data set calls that CA Telon generates.

N

CA Telon generates the 01 and 02 levels and uses the COPY or INCLUDE member for 03 levels and below.

COPYLBL

The COBOL or PL/I data item name of the group level (for example, an 01 field and its subordinates) for the segment copy definition. For example:

01 TRG1-IO-AREA

This value overrides the default I/O area *IOA-record-name-SEGMENT*.

If two records have the same record name (the default value of DBSEG or some other name that you specify in the LABEL field) and the same COPYLBL value, CA Telon generates only one I/O area. Thus, for CA Telon to generate a second I/O area, specify a unique data item name in the COPYLBL field.

If you enter **Y** in the COPYLV1 field and CA Telon is generating automatic I/O, a COPYLBL value is required.

COBDIV

The section of the COBOL program in which to copy the record layout for the data set. The definition is to begin at level 03 or higher and will have a 02 level of 02 IOA-*record-name*-SEGMENT generated above it.

If you do not specify a value, CA Telon copies the record layout into the COBOL file definition for the data set. If you code Working Storage, the record layout appears in the IO-AREA of the program. This value in this field is valid only on batch programs and on RECORD statements referencing a data set. See the *Programming Concepts Guide* for information on RECORD statements.

COBVSKEY

The data name of the VSAM KSDS or RRDS key for the file.

For a VSAM KSDS, you must define the key for the file in the KEY field. For a VSAM RRDS, the data name must specify an unsigned integer defined in the Working Storage of the COBOL program.

Note: Since CA Telon does not generate the relative key, you must define it.

COBOL uses this variable-keyed access to VSAM files. The value in this field is valid only in batch programs and on RECORD statements referencing a VSAM key-sequenced or relative data set.

To specify that the KSDS alternate key (key on a data set using the INDEXOF field value) does not have to be a unique data set, enter **DUPLICATE** after the data name.

KEY

The PL/I or COBOL variable names containing the key to a record. If inheritance is requested, data access (user I/O) uses the variable name to identify a record that it retrieves.

See the *Programming Concepts Guide* for information on inheritance.

KEYLEN

The length of the RID (Record Identification) field.

OPCODE

(DL/I processing.) The OPCODE value for VSAM access. Default values are **>=** for BROWSE segments and **=** for all others.

TYPE

The type of access for the data set. Values are:

KSDS

(Default.) VSAM key-sequenced data set

RRDS

VSAM relative data set

RELATIVE

VSAM relative data set

ESDS

VSAM entry-sequenced data set

ACCMODE

Access to this type of data set. Values are:

DYN

(Default) VSAM dynamic access

RAN

VSAM random access

SEQ

VSAM sequential access

DIR

VSAM direct access

OPTLIST

(*VSAM processing only.*) Options on CICS data sets for command level calls. Values are:

- RRN
- SEGSET
- SEGSETALL
- SYSID
- MASSINSERT
- DEBKEY
- DEBREC
- UPDATE

You can specify one or more of these options on the SEGMENT and all user exec data access (READ, UPDATE, CREATE, and DELETE). Separate each option with a comma.

Values that you code on the SEGMENT statement are carried down to user exec specifications if you do not specify an OPTLIST value on the user exec data access. Values that you specify that are not valid for a particular command level verb (for example, READ, WRITE, and STARTBR) are automatically removed from any calls using that verb. For example, if MASSINSERT is specified as an option on the RECORD statement or user exec data access, it appears only if the EXEC CICS WRITE command is generated for that segment in the CICS program.

You can specify literal values for the SEGSET and SYSID options by enclosing the literal value in double quotes. For example:

```
OPTLIST=(RRN,SYSID("SYSA"))
```

RECLTH

(VSAM processing of variable-length records only.) The maximum length of each record on the file, as follows:

Record-length

Maximum record length, specified as either an integer or the name of a COBOL or PL/I variable that contains the key-length value. This value is used when reading or writing the VSAM record. Any rewrite operations are processed using the current length of the record being updated (as determined by the read).

Read-length, rewrite-length

Maximum record length during a read and subsequent update, respectively. Each value can be specified as an integer or the name of a COBOL or PL/I variable that contains the key-length value. Read-length is used for all automatic read EXEC CICS calls for the record.

Rewrite-length is used as the maximum length of the updated record.

This specification is applicable for UPDATE processing only (for example, usage is UPDATE).

If you do not enter a value for this field, CA Telon uses the value from the RECORD statement.

Note: The length that you use in a read operation for a variable-length record must be at least as large as the actual record retrieved. If not, a CICS AEIV (LENGERR) ABEND results.

GENKEYL

(VSAM processing only.) The length of the generic key used for the access. This value can be an integer or the name of a COBOL or PL/I variable that contains the key-length value.

If you do not specify a value, CA Telon assumes that the access uses the full key length; it does not use the value from the RECORD statement.

INDEXOF

A value to specify that this data set is the index of another data set. It must directly follow the indexed data set on the Create/Update SSA/Command For DL/I DB/Segment screen.

REUSE

A value to specify whether I/O should start at the beginning or end of a data set that is opened to output.

Values are:

Y

I/O starts at the beginning of the data set

N

I/O starts at the end of the data set if any data already exists

Update CICS Queue Default Data

Access

On the Data Administration menu, enter:

- **CR** or **UP** in the FUNCTION field
- **CQ** in the ITEM field
- *Name* in the NAME field
- *Description* in the DESCRIPTION field

Program ID

D11Q

Function

Collects data for a queue's record definition at a global (data administration) level. Queues defined here can subsequently be added to the data groups of CICS screen and nonterminal definitions with the DGADD command.

Note: Queues that are not defined in data administration can be created directly in a data group by using the DGADD command. They are added with skeletal information which you can then modify.

```

UPDATE CICS QUEUE DEFAULT DATA *****
COMMAND=>
CQNAME *****
GENERAL: TYPE    ___ (TS/TD)    AUX/MAIN  (A/M)
*        LRECL   ___
*        SYSID   _____
RECORD:  LABEL   _____
*        COPY    _____
*        COPYLV1 (Y/N)        COPYLBL  _____
  
```

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

CQNAME

(*Protected field.*) Displays the default queue name to identify this queue to CICS. The value in this field was specified in the NAME field on the Data Administration menu.

This value is the default for all program definitions that reference this queue. You can override it at the data group record level by using the variable QUELBL. You cannot override it at the program definition level (that is, on the Update CICS Queue Record screen).

TYPE

The type of CICS queue. Values are:

TS

Temporary storage

TD

Transient data

This value is the default for all program definitions that reference this queue. You cannot override it at the program definition level.

AUX/MAIN

(Temporary storage queues only.) The type of storage to which this queue should be written. Values are:

A

(*Default.*) Auxiliary storage

M

Main storage

This value is the default for all program definitions that reference this queue. You cannot override it at the program definition level.

LRECL

The length (in bytes) of this queue's entries. This value is required if the LTHOPT value specified on the Update CICS Queue Record screen is Y and the value in the TYPE field is TS.

This value is the default for all program definitions that reference this queue. You can override it at the program definition level on the Update CICS Queue Record screen.

SYSID

The system ID to use when CA Telon accesses this queue.

A value is not required, but if you specify one, you must also specify LTHOPT and LRECL values on the Update CICS Queue Record screen.

This value is the default for all program definitions that reference this queue. You cannot modify it at the program definition level.

The variable loaded with SYSID is one of the following:

- SYSWK-cqname-QUEUE-SYSID (COBOL) or
- SYSWK_cqname_QUEUE_SYSID (PL/I), where cqname is the value of the CQNAME field.

LABEL

A replacement for the name of the CICS queue in the user exec (U-100) paragraph name that CA Telon generates for this queue.

This value is the default for all program definitions that reference this queue. You can override it at the program definition level on the Update CICS Queue Record screen.

COPY

The COBOL COPY or PL/I INCLUDE member name that contains the queue record layout.

If you do not specify a value, programs that access this queue will COPY/INCLUDE a member using the CQNAME value. If you enter **NONE**, CA Telon does not generate a COPY or INCLUDE statement for this queue.

This value is inherited by all program definitions that reference this queue. You can override it at the program definition level on the Update CICS Queue Record screen.

COPYLV1

A value to specify whether the COBOL COPY or PL/I INCLUDE member that defines the layout of this queue record begins with a 01-level data item. Values are:

Y

The COPY or INCLUDE begins with a 01-level data item (for example, 01 TRG1-IO-AREA)

N

(*Default.*) The COPY or INCLUDE begins with a data item of level 03 or higher

This value is inherited by all program definitions that reference this queue. You can override it at the program definition level on the Update CICS Queue Record screen.

COPYLBL

The COBOL or PL/I group-level variable name for the CICS queue accesses. This value overrides the FROM/INTO data area used in all I/O generated for this CICS queue. For example, if the data area for this queue begins with the COBOL variable 01 TRG1-IO-AREA, the COPYLBL value must be TRG1-IO-AREA.

This value is inherited by all program definitions that reference this queue. You can override it at the program definition level on the Update CICS Queue Record screen.

Update CICS Journal Default Data

Access

On the Data Administration menu, enter:

- **CR** or **UP** in the FUNCTION field
- **CJ** in the ITEM field
- *Name* in the NAME field
- *Description* in the DESCRIPTION field

Program ID

D11J

Function

Collects data for a journal's record definition at a global (data administration) level. Journals defined here can subsequently be added into the data groups of CICS screen and nonterminal definitions with the DGADD command.

Note: Journals not defined in data administration can also be created directly in a data group by using the DGADD command. Journals are created with default initial values which you can then modify.

```

UPDATE CICS JOURNAL DEFAULT DATA **** *****
COMMAND ==> _____
CJNAME _____ *****
GENERAL: JFILEID __ (01 - 99)
        * JTYPEID __
        * LRECL  __
RECORD: LABEL _____
        * COPY   _____
        * COPYLV1 (Y/N)      COPYLBL _____
  
```

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

CJNAME

(Protected field.) Displays the name of this CICS journal definition. The value in this field was specified in the NAME field on the Data Administration menu.

This value is the default in all program definitions that reference this CICS journal. You cannot override it at the program definition level.

Note: If the journal is not defined in data administration, you can specify during DGADD any CJNAME value at the program level that is not defined in TDF data administration.

JFILEID

Journal file ID. Values in CICS are 02 through 99. (01 is reserved by CICS for the system log.)

This value is the default in the program definitions that reference this CICS journal. You cannot modify it at the program level if the journal is defined to data administration.

JTYPEID

Two characters used by CICS to identify the origin of this journal record.

This value is the default in the program definitions that reference this CICS journal. You cannot modify it at the program level if the journal is defined to data administration.

LRECL

The maximum length (in bytes) of the user data in this journal's entries. The total length is the sum of the prefix length (if any) and the data length. The Generator assigns this value to the default journal length variable, *SYSWK-journal-name-JOURNAL-LENGTH*, or its user-defined override.

This value is the default in the program definitions that reference this CICS journal. You can modify it at the program level on the Update CICS Journal Record screen.

LABEL

A replacement for the CJNAME value in the user exec (U-100) paragraph names that CA Telon generates for this journal.

This value is the default in the program definitions that reference this CICS journal. You can modify it at the program level on the Update CICS Journal Record screen.

COPY

The COBOL COPY or PL/I INCLUDE member name that contains the journal record layout. If you do not specify a value, programs that access this journal use the value in the CJNAME field in a COPY or INCLUDE statement. If you enter **NONE**, CA Telon does not generate a COPY or INCLUDE statement.

This value is inherited by all program definitions that reference this CICS journal. You can modify it at the program level on the Update CICS Journal Record screen.

COPYLV1

A value to specify whether the COBOL COPY or PL/I INCLUDE member that defines the layout of this journal record begins with a 01-level data item. Values are:

Y

The COPY or INCLUDE begins with a 01-level data item (for example, 01 TRG1-IO-AREA)

N

(*Default.*) The COPY or INCLUDE begins with a data item of level 03 or higher

This value is inherited by all program definitions that reference this queue. You can override it at the program definition level on the Update CICS Journal Record screen.

COPYLBL

The COBOL or PL/I group-level variable name for the CICS journal accesses. This value overrides the FROM/TO data area used in all I/O generated for this CICS journal.

For example, if the data area for this journal begins with the COBOL variable 01 JOURNAL-SAVE, the COPYLBL value must be JOURNAL-SAVE.

This value is inherited by all program definitions that reference this CICS journal. You can override it at the program definition level on the Update CICS Journal Record screen.

Catalog/Import DB2 Tables

Access

On the Data Administration menu, enter:

- **CA** in the FUNCTION field
- **D2** in the ITEM field
- *Name* in the NAME field
- *Description* in the DESCRIPTION field

Program ID

D411

Function

Imports DB2 tables or views into the TDF.

For more information about importing, see the *Utilities Guide*.

XXXXXX

CATALOG/IMPORT DB2 TABLES *****

COMMAND ==>

PAGE 01

SELECT THE DB2 TABLES TO BE IMPORTED INTO THE TDF

QUAL ****TABLENAME***** COLS TLN TLNNAME USERID UPDATE

A

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

(Type of import) A

The type of import to perform. Values are:

I

Import into the TDF only if the table was not previously imported. The message ***DUP - USE O/A** appears if you try to import a previously imported table.

O

Import into the TDF whether or not the table was previously defined to the TDF. If the table exists in the TDF, overlay existing data with imported data.

A

Import this table's catalog information as one or more additional TLNROWS to the table specified in the TLNNAME field. If the TLNNAME is blank, this table's current catalog information is added as additional TLNROW to its existing TDF information.

QUAL

(Protected field.) Identifies the table name qualifier.

TABlename

(Protected field.) Identifies the name of the table. CA Telon extracts this table from the catalog.

COLS

(Protected field.) Identifies the number of columns in the table. CA Telon extracts this table from the catalog.

TLN

(Protected field.) Indicates whether the table has been imported into the TDF. Possible values are:

Y

The table has been imported

N

The table was not imported

TLNNAME

The unique CA Telon name for the table or join used by the TDF and Generator to construct host variable names. The default is the first eight characters of the table name.

On an ADD request, this value specifies the table being extended.

USERID

(Protected field.) Identifies the user who imported the table.

UPDATE

(Protected field.) Identifies the date of the most recent update or import.

(Message area)

(Protected field.) Displays any error condition that occurs during an import or displaying ***IMPORT OK** when the import is successful.

List SQL Tables/Joins

Access

On the Data Administration menu, enter:

- **LI** in the FUNCTION field
- **TB** in the ITEM field to list tables, TJ to list joins
- *Name* in the NAME field
- *Description* in the DESCRIPTION field

Program ID

D402

Function

Lists SQL tables and views or SQL joins that have been imported into the TDF.

```
LIST SQL TABLES *****
COMMAND ==> _____ PAGE 01
** *****SQL QUAL.NAME***** *****DESCRIPTION***** *TLNNAME* DB
A _____
```

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

(Select field) A

Use this column to enter these commands:

C

Copy

R

Rename

U

Update

Z

Zap (unconfirmed delete)

Note: You may zap multiple tables at one time. The other commands can be performed on only one table at a time.

SQL QUAL.NAME

(*Protected field.*) Identifies the qualifier and table name.

DESCRIPTION

(*Protected field.*) Displays a description of the table.

TLNNAME

(*Protected field.*) Identifies the unique name for the table or join used by CA Telon to construct the names of host variables that it generates.

DB

(*Protected field.*) Indicates the SQL type. These are the following values:

D2

DB2 family (DB2, SQL/400, SQL/DS, XDB)

DC

CA-Datacom/SQL

ID

CA-IDMS/SQL

Create/Update SQL Tables/TLNROWS/Temporary Tables

Access

On the Data Administration menu, enter:

- **CR** or **UP** in the FUNCTION field
- **TB** in the ITEM field
- *Name* in the NAME field
- *Description* in the DESCRIPTION field

You can also reach these screens by entering **U** as a line command on the List SQL Tables/Joins screen.

Program ID

- D141 (SQLs other than CA-IDMS or CA-Datacom/SQL)
- D151 (CA-IDMS/SQL or CA-Datacom/SQL, discussed in the section following this one)

Function

Appends CA Telon fields to an existing SQL table definition that has been imported from the SQL catalog into the TDF.

Note: If you create a table, the table is defined only in the TDF. All SQL tables must be created in the host SQL environment.

You can then specify additional parameters to enhance CA Telon's capability to generate the COBOL or PL/I code that you want. Use this screen to specify these additional parameters.

The Create/Update SQL Table/TLNROWS screen for SQLs other than CA-IDMS or CA-Datacom is shown following. QUAL.TBLNAME refers to the full table name, including qualifier (QUAL) and name (NAME). SIZE indicates the total number of lines contained in the list of column names.

UPDATE SQL TABLE *****		*****		SIZE 000010
COMMAND ==>				SCROLL ==>
TLNNAME	DESCR			SYNONYM Y/N
DCLCOPY	DCLLBL			DCLRDEF Y/N
COPY	COPYLBL			COPYLV1 Y/N

A	COLUMN NAME	ALIAS	KY/AC	TYPE	LTH	DC	^NU
*****	*****	*****	*****	*****	*****	*****	*****
		TOP OF DATA	*****	*****	*****	*****	*****
000001	--TLNROW--						
000002							
000003							
000004							
000005							
000006	--TLNROW--			*TMP			
000007							
000008							
000009							
000010							
*****	*****	*****	*****	*****	*****	*****	*****
		BOTTOM OF DATA	*****	*****	*****	*****	*****

Show/Purge screen

You can access the Show/Purge SQL Tables/TLNROWS (D147) screen, or the Show/Purge CA-IDMS/CA-Datacom SQL Tables/TLNROWS (D157) screen from the Data Administration menu by entering:

- **SH** or **PU** in the FUNCTION field
- **TB** in the item field
- **Name** in the NAME field

Alternately, on the List SQL Tables/Joins screen (D402), you can enter S or P as a line command for the desired table.

Field Definitions

The Show/Purge screen fields are the same as those found in the Create/Update screens for these tables, except that they are all protected in the Show/Purge screen.

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

TLNNAME

The CA Telon name for the table, used to construct the names generated for such host variables as paging keys and not-null indicators.

The default value is the value in the DCLCOPY field.

The TLNNAME must be unique for all Tables in the TDF regardless of the type (SQL or CA-IDMS/SQL or CA-Datcom/SQL) of table.

This parameter can be the same name as the name you specify for the DCLCOPY parameter, described later in this section. You should use a different name for the TLNNAME if the DCLCOPY name is encoded and you want a more meaningful name for the variable names that CA Telon generates.

DESCR

A catalog description of the table, for informational purposes only. If you do not specify a value in this field, CA Telon displays the default description provided by the import method.

SYNONYM

(SQL Table) Specifies whether CA Telon is to generate the qualifier for the table. Values are:

Y

CA Telon leaves the qualifier for the table blank, with appropriate processing based on the synonyms specified outside the TDF environment

N

CA Telon generates the qualifier for the table

DCLCOPY

The name that CA Telon gives in an EXEC SQL INCLUDE. For DB2, it is the SQL DCLGEN COPY member name. If you are always going to specify the INCLUDE in custom code work areas, enter **NONE** in this field.

CA Telon requires you to enter a value in this field.

DCLLBL

The name for the COBOL or PL/I structure that the DCLGEN produces for this table. If you do not enter a value, CA Telon uses the default structure name (the table name is prefixed by DCL).

DCLRDEF

A value to specify whether CA Telon is to redefine the DCLGEN INCLUDE structure with the COPY or INCLUDE member that you specify in the COPY field. Values are:

Y

Redefine the DCLGEN INCLUDE structure with the COPY member that you specify.

N

(Default) Do not redefine the DCLGEN INCLUDE structure. CA Telon copies the COPY or INCLUDE member into Working Storage.

COPY

The COPY/INCLUDE member that CA Telon is to copy into Working Storage for this table at generation time.

This COPY or INCLUDE member name is the member name for the ALIASes. For DB2 Tables, it is not a DB2 INCLUDE member.

To prevent the copy, enter **NONE**.

COPYLBL

The COBOL data item name of the group level for the Table I/O area copy definition (for example, 01 TRG-IO-AREA). The default I/O area in all table calls is IOA-TLNNAME-SEGMENT.

See Update DBD Segment for more information on COPYLBL.

COPYLV1

A value that specifies whether the COBOL COPY or PL/I INCLUDE member identified in the COPY field begins with a 01-level data item. Values are:

Y

The COPY or INCLUDE begins with a 01-level data item (for example, 01 TRG1-IO-AREA)

N

(Default.) The COPY or INCLUDE begins with a data item of level 03

(Line command) A

A field where you can enter one of the following line commands:

C

Copy a line

CC

Copy a block of lines

I

Insert a line

Inn

Insert *nn* lines

IS(*n*)

Insert *n* lines for you to enter the command

M

Move a line

MM

Move a block of lines

R

Repeat a line

RR

Repeat a block of lines

A

Line(s) to insert, copy, or move go after this line

B

Line(s) to insert, copy, or move go before this line

D

Delete an entry.

See Line Commands for more information.

COLUMN NAME

Identifies the name of the column in the table or the start of a TLNROW (a row in a CA Telon-imported table) which redefines the columns of the table. You can insert a TLNROW by entering a hyphen (-) for this parameter.

Although you can use this parameter to specify the name of a column in the table, to minimize keying errors, CA recommends that you copy the columns for a TLNROW from the original set of columns.

If desired, you can delete extra columns from the TLNROW.

The literal —TLNROW-- identifies a TLNROW line. The TLNROW name is the value in the ALIAS field for SQL tables. CA Telon requires that the name of the first TLNROW match the TLNNAME value.

ALIAS

This field can contain one of the following:

- An I/O area or host variable name for the column. CA Telon uses this I/O area or host variable name in place of the column name for the generated I/O.
- The name of the TLNROW that you are defining. CA Telon uses the name of a TLNROW in the generated I/O procedure or paragraph names.
- An SQL Special register: CURRENT TIME, CURRENT DATE, or CURRENT TIMESTAMP.

KY

A value to specify whether this column is involved in generated SQL I/O WHERE conditions. Values are:

Y

Declares this column a key column. CA Telon includes any such columns in sequential order in the generated WHERE condition.

P

Generates the WHERE condition to include this column in position *p* of the column order. If a number is specified for one column, Y is not valid for any column. The group of numbers must begin with 1 and be incremented by 1.

AC

A value to specify whether the EXEC SQL statement is to access the column. Values are:

Y

The column is accessed by the EXEC SQL statement

N

The column is not accessed by the EXEC SQL statement

TYPE (TLNROW)

To identify a TLNROW as a temporary table, enter ***TMP** in the TYPE field for TLNROW.

TYPE (COLUMN)

The type of fields in the table. Values are listed in the following table:

SQL Data Type	CA Telon Field			Description
	TYPE	LTH	DEC	
CHAR (<i>n</i>)	CHAR	<i>n</i>		A fixed-length character string of length <i>n</i> where <i>n</i> is in the range 1-254.
DASH				A TLNROW.
DATE	DATE			A date value in either of two forms: U.S. or International.
DECIMAL (<i>p, s</i>)	DEC	<i>p</i>	<i>s</i>	A decimal number. The first integer, <i>p</i> , is the precision of the number (the total number of digits) and must be in the range 1 to 15. The second integer, <i>s</i> , is the scale of the number (the number of digits to the right of the decimal point) and must be in the range 0 to <i>p</i> .
FLOAT	FL			A floating point number.
GRAPHIC (<i>n</i>)	GR	<i>n</i>		A fixed-length string of double-byte characters of length <i>n</i> where <i>n</i> is in the range 1-127.
INTEGER	INT			A long integer.
LONG VARCHAR	LVCH			A varying-length character string whose maximum length is determined by the host SQL.
LONG VARGRAPHIC	LVGR			A varying-length character string whose maximum length is determined by the host SQL.
SMALLINT	SINT			A small integer.
TIME	TIME			A time value.
TIMESTAMP	STMP			The timestamp.
VARCHAR (<i>n</i>)	VCHR	<i>n</i>		A varying-length character string of maximum length <i>n</i> that must be in the range 1 to 32,767.
VARGRAPHIC (<i>n</i>)	VGR	<i>n</i>		A varying-length string of double-byte

characters of maximum length n that must be in the range 1 to 32,767.

LTH

The length, if the type is variable length, or the precision, if the type is DECIMAL. See the table in the TYPE field documentation.

If you are creating an SQL table, enter a value. If you are updating a table, CA Telon displays the value.

For updates and imports, CA Telon obtains this field from the SQL catalog. If you update the value here, you must update the corresponding value in the catalog.

DEC

The scale, if the type is DECIMAL. See the table in the TYPE field documentation.

If you are creating a table, enter a value. If you are updating a table, CA Telon displays the value.

For updates and imports, CA Telon obtains this field from the SQL catalog. If you update the value of this field for SQL tables, you must update the corresponding value in the catalog.

^N (NOTNULL)

A value to specify that null values are not allowed in the column. Values are:

Y

Nulls not allowed.

N

Nulls allowed. CA Telon generates null indicator variables associated with each column defined with N in this field.

If you are creating a table, enter a value. If you are updating a table, CA Telon displays the value.

For imports, CA Telon obtains this field from the SQL catalog. If you update the value here, you must update the corresponding value in the catalog.

TYPE (TLNROW)

To identify a TLNROW as a temporary table, enter *TMP.

Update CA IDMS/SQL CA Datacom/SQL Tables/TLNROWs

Access

On the Data Administration menu enter:

- **UP** in the FUNCTION field
- **TB** in the ITEM field
- *Name* in the NAME field

You can also reach this screen by entering **U** as a line command on the List SQL Tables/Joins screen.

Program ID

- D151 (CA IDMS/SQL or CA Datacom/SQL)

Function

This screen is used to append CA Telon table rows to an existing CA IDMS/SQL or CA Datacom/SQL table definition.

You do not create a CA Telon definition for CA IDMS/SQL or CA Datacom/SQL table within the TDF as you do for other SQL implementations. (See the Create/Update SQL Table screen (D141) described in the previous section.) To ensure that the TDF and catalog information are the same, definitions for CA IDMS/SQL and CA Datacom/SQL tables must be extracted from the appropriate catalog and transported into the TDF using the CA Telon Transport utility. See the *Advantage CA IDMS® Database SQL Option Guide* and *Advantage CA Datacom® Database SQL Option Guide* manuals for the complete details on this process.

Once the SQL table definition has been transported in, you can copy the primary TLNROW to create view TLNROWS to enhance CA Telon's capability to generate the COBOL code that you want. Use this screen to specify this information.

The Update SQL Table/TLNROWS screen for CA IDMS/SQL or CA Datacom/SQL is shown next.

In the next screen display, QUAL.TBLNAME refers to the full table name, including qualifier (QUAL) and name (NAME). SIZE indicates the total number of lines contained in the list of column names.

UPDATE SQL TABLE ***** QUAL.TBLNAME ***** SIZE 000000																						
COMMAND ==> _____	SCROLL ==> _____																					
TLNNNAME _____	DESCR _____ TYPE _____																					
COPY _____	COPYLBL _____ COPYLVL _____ Y/N																					
DICTNAM _____	SCHEMA _____ SYNONYM _____ Y/N																					
<table border="1"> <thead> <tr> <th>COLUMN NAME</th> <th>ROW NAME</th> <th>ALIAS</th> <th>KY/AC</th> <th>TYPE</th> <th>LTH/DEC</th> <th>^N</th> </tr> </thead> <tbody> <tr> <td>***** ***** TOP OF DATA *****</td> <td>*****</td> <td>*</td> <td>** *</td> <td>*****</td> <td>****</td> <td>** *</td> </tr> <tr> <td>A _____</td> <td>_____</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> </tbody> </table>		COLUMN NAME	ROW NAME	ALIAS	KY/AC	TYPE	LTH/DEC	^N	***** ***** TOP OF DATA *****	*****	*	** *	*****	****	** *	A _____	_____	-	-	-	-	-
COLUMN NAME	ROW NAME	ALIAS	KY/AC	TYPE	LTH/DEC	^N																
***** ***** TOP OF DATA *****	*****	*	** *	*****	****	** *																
A _____	_____	-	-	-	-	-																

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

TLNNNAME

The CA Telon name for the table, used to construct the names generated for such host variables as paging keys and not-null indicators.

The TLNNNAME must be unique for all tables in the TDF regardless of the type (DB2, CA IDMS/SQL, CA Datacom/SQL, and so forth) of table.

Note: When an SQL table definition is extracted from a dictionary, TLNNNAME defaults to the first eight characters of the table name if you do not specify a more meaningful TLNNNAME in the control cards.

DESCR

A catalog description of the table, for informational purposes only. If you do not specify a value in this field, CA Telon displays the default description provided by the extract utility.

TYPE

Identifies the type of SQL Table:

- BASE
- VIEW
- NETWORK (CA IDMS/SQL only)

COPY

The COPY/INCLUDE member that CA Telon is to copy into Working Storage for this table at generation time.

This COPY or INCLUDE member name is the member name for the ALIASes.

To prevent the copy, enter **NONE**.

COPYLBL

The COBOL data item name of the group level for the Table I/O area copy definition (for example, 01 TRG-IO-AREA). The default I/O area in all table calls is IOA-TLNNAME-SEGMENT.

For more information about COPYLBL, see Update DBD Segment.

COPYLV1

A value that specifies whether the COBOL COPY member identified in the COPY field begins with a 01-level data item. Values are:

Y

The COPY begins with a 01-level data item (for example, 01 TRG1-IO-AREA)

N

(Default.) The COPY begins with a data item of level 03

DICTNAM

Displays the CA IDMS Dictionary Name (DICTNAME) from which the table definition was extracted (CA IDMS/SQL only).

SCHEMA

Displays the schema name of the table.

SYNONYM

Specifies whether CA Telon is to generate the qualifier for the table (CA Datacom/SQL table only). Values are:

Y

CA Telon leaves the qualifier for the table blank, with appropriate processing based on the synonyms specified outside the TDF environment

N

CA Telon generates the qualifier for the table

(Line command) A

A field where you can enter one of the following line commands:

C

Copy a line

CC

Copy a block of lines

I

Insert a line

Inn

Insert nn lines

IS(n)

Insert n lines for you to enter the command

M

Move a line

MM

Move a block of lines

R

Repeat a line

RR

Repeat a block of lines

A

Line(s) to insert, copy, or move go after this line

B

Line(s) to insert, copy, or move go before this line

D

Delete an entry

U

Update a TLNROW

S

Select a TLNROW

The valid line commands depend on the type of line for which the command is entered. The primary TLNROW is defined as the principal Table definition imported from the dictionary. A primary TLNROW cannot be modified. A view TLNROW is defined as a redefinition, or view, of the principal Table definition and is created and maintained by the user.

For the primary TLNROW, only the U(pdate) and S(lect) line commands may be entered.

For a primary TLNROW column, only the following line commands may be entered:

C

Copy a line

CC

Copy a block of lines

I

Insert a line after the last primary Table column

Inn

Insert nn lines after the last primary Table column

For a view TLNROW, all line commands are valid.

For a view TLNROW Column, all line commands except U(pdate) and S(lect) are valid.

See Line Commands for more information.

COLUMN NAME

Identifies the name of the column in the table or the start of a TLNROW (a row in a CA Telon imported table) which redefines the columns of the table.

For view TLNROWS, you must copy the columns from the original set of columns.

For SQL table primary TLNROW columns, this field cannot be specified.

The literal —TLNROW-- identifies a TLNROW line. The TLNROW name is the value in the ROW NAME field for SQL Tables. CA Telon requires that the name of the first, or primary, TLNROW match the TLNNAME value for the table.

ROW NAME

This field provides the name of the TLNROW you are defining. CA Telon uses the name of the TLNROW in the generated I/O procedure or paragraph names.

ALIAS

This field indicates whether an ALIAS has been established for the columns on the Update SQL ALIAS screen. Values are:

Y

Yes

N

No

Note: To view or maintain SQL table column aliases, enter a U(pdate) or S(elect) in the line command field of the TLNROW. Aliases are not used for CA Datacom.

KY

A value to specify whether this column is involved in generated SQL I/O WHERE conditions. Values are:

Y

Declares this column a key column. CA Telon includes any such columns in sequential order in the generated WHERE condition.

P

Generates the WHERE condition to include this column in position *p* of the column order. If a number is specified for one column, Y is not valid for any column. The group of numbers must begin with 1 and be incremented by 1.

AC

A value to specify whether the EXEC SQL statement is to access the column. Values are:

Y

The column is accessed by the EXEC SQL statement

N

The column is not accessed by the EXEC SQL statement

TYPE

The type of fields in the table. Values are listed in the following table:

IDMS/SQL Data Type	CA Telon Field	Description of Data Type		
	TYPE	LTH	DEC	
CHAR (<i>n</i>)	CHAR	<i>n</i>		A fixed-length character string of length <i>n</i> where <i>n</i> is in the range 1-254.
DATE	DATE			A date value in either of two forms: U.S. or International.
DECIMAL (<i>p</i> , <i>s</i>)	DEC	<i>p</i>	<i>s</i>	A decimal number. The first integer, <i>p</i> , is the precision of the number (the total number of digits) and must be in the range 1 to 15. The second integer, <i>s</i> , is the scale of the number (the number of digits to the right of the decimal point) and must be in the range 0 to <i>p</i> .
DOUBLE PRECISION	DBLPREC			A 64-bit (long) floating-point value with a seven-bit exponent and a binary precision of 56. The length of a DOUBLE PRECISION value is eight bytes.
FLOAT	FL			A floating point number.
GRAPHIC (<i>n</i>)	GR	<i>n</i>		A fixed-length string of double-byte characters of length <i>n</i> where <i>n</i> is in the range 1-127.
INTEGER	INT			A long integer.
LONG INTEGER	LONGINT			A 64-bit signed long integer value. The length of a LONG INTEGER value is eight bytes (CA IDMS/SQL only).
NUMERIC (<i>p</i> , <i>s</i>)	NUMERIC	<i>p</i>	<i>s</i>	A fixed-point, signed zoned decimal value. The first integer, <i>p</i> , is the precision of the number (the total number of digits) in the range 1 to 32. The second integer, <i>s</i> , is the scale of the number of digits to the right of the decimal point) in the range 0 to <i>p</i> .
REAL	REAL			A 32-bit (short) floating-point value with a seven-bit exponent and a binary precision of 24. The length of a REAL value is four bytes.
CA IDMS/SQL Data Type	CA Telon Field	Description of Data Type		
	TYPE	LTH	DEC	

SMALLINT	SINT		A small integer.
TIME	TIME		A time value.
TIMESTAMP	STMP		The timestamp.
VARCHAR (n)	VCHR	n	A varying-length character string of maximum length n that must be in the range 1 to 32,767.
VARGRAPHIC (n)	VGR	n	A varying-length string of double-byte characters of maximum length n that must be in the range 1 to 32,767.

LTH

(*Protected field.*) Displays the length, if the type is variable length, or the precision, if the type is DECIMAL, NUMERIC, UNSIGNED DECIMAL, or UNSIGNED NUMERIC. See the table in the TYPE field documentation.

For imports, CA Telon obtains this field from the dictionary.

DEC

(*Protected field.*) Displays the scale, if the type is DECIMAL, NUMERIC, UNSIGNED DECIMAL, or UNSIGNED NUMERIC. See the table in the TYPE field documentation.

For imports, CA Telon obtains this field from the dictionary.

^N (NOTNULL)

A value to specify that null values are not allowed in the column. Values are:

Y

Nulls not allowed

N

Nulls allowed. CA Telon generates null indicator variables associated with each column defined with N in this field.

For imports, CA Telon obtains this field from the dictionary.

Update SQL ALIAS

Access

Access this screen by entering a U(pdate) or S(elect) in the line command field of the TLNROW line on the Update CA-IDMS/CA-Datacom/SQL Tables/TLNROWS screen.

Program ID

D152

Function

This screen allows maintenance of a host variable name for each column name within the TLNROW. CA Telon uses this host variable name in place of the column name for the generated I/O.

The Update SQL ALIAS screen is displayed with differing content depending upon the SQL. For CA-IDMS/SQL, the screen appears as follows:

```

***** ADD/UPDATE SQL ALIAS *****  ***  QUAL.TBLNAME *****
COMMAND ==> _____ SCROLL ==> _____
TLNROW _____ DESCR _____
COPY _____ COPYLBL _____ COPYLV1
DICTNAM _____ SCHEMA _____

          COLUMN NAME                      ALIAS
          *****                      *****
          _____                      _____

```

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

TLNROW

Identifies the TLNROW name of the listed columns.

DESCR

Displays the catalog description of the table.

COPY

Displays the name of the COPY or INCLUDE member for the table or NONE.

COPYLBL

Displays the COBOL data item name of the group level for the table I/O area copy definitions, if specified on the Update CA-IDMS/CA-Datcom/SQL Table screen.

COPYLV1

Displays the COBOL COPY Level 01 indicator as specified on the Update CA-IDMS/CA-Datcom/SQL Table screen.

DICTNAM

Displays the CA-IDMS Dictionary Name from which the table definition was extracted.

SCHEMA

Displays the schema name of the table.

COLUMN NAME

Identifies the name of the column in the TLNROW.

ALIAS

This field can contain the host variable name for the column used by CA Telon in place of the column name in the generated I/O.

Note: All the fields are protected except COMMAND and ALIAS.

Specify Tables Being Joined

Access

On the Data Administration menu, enter:

- **CR** or **UP** in the FUNCTION field
- **TJ** in the ITEM field
- *Name* in the NAME field
- *Description* in the DESCRIPTION field

Program ID

D142

Function

Specifies the tables involved in join processing.

SPECIFY TABLES BEING JOINED *****

COMMAND ==>

JOIN NAME

TLNNAME

	CORRELATION				SQL	
A	NAME	QUAL	TABLE NAME	TLNNAME	TYPE	SQL SCHEMA
	*****	*****	*****	*****	*****	*****
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

JOIN NAME

Qualified join name.

TLNNAME

The CA Telon name for the join. This name must be unique among join or table identifiers.

(OUTPUT LINE) A

(Protected field.) Displays the table sequence number.

CORRELATION NAME

A correlation name for the table in the generated auto exec (user I/O) for this join.

QUAL

The qualifier of the table used in this join.

TABLE NAME

The name of a table used in this join.

TLNNAME

The CA Telon name of the table used for this join. CA Telon uses this value to construct the names of such host variables as paging keys and not-null indicators.

SQL TYPE

(Protected field.) Displays the type of SQL table. Values are DB2, IDMS, and DATACOM. All tables used in the join must be the same SQL type.

SQL SCHEMA

This field is displayed only for CA-IDMS/SQL or CA-Datcom/SQL Tables in the join and is a protected field identifying the CA-IDMS/SQL or CA-Datcom/SQL Schema name of the table.

Note: For UPDATE, a change in the CORRELATION NAME, QUAL, or TABLE NAME is processed as a delete of that table from the join and add of a new table.

Update SQL Join— Join Columns

Access

On the Specify Tables Being Joined screen, invoke END processing.

Program ID

D143

Function

Specifies the WHERE clause predicate(s) used in constructing the join part of the WHERE clause in generated user I/O.

CA Telon constructs the join WHERE clause predicates by using the column names that you specify, as well as the correlation name of the table that defines the column. If you specify more than one predicate, CA Telon constructs the entire WHERE clause by joining the predicates with the AND keyword.

If a table is added to an existing join, the display of the join columns is followed by the columns of the newly added table. If a table is deleted from a join on the previous screen, columns from the tables it was joined to will follow the display of any remaining joined columns.

```
UPDATE SQL JOIN - JOIN COLUMNS ***** QUAL.TBLNAME ***** SIZE 000000
COMMAND ==> _____ SCROLL ==> ____
ACCESS COLUMNS _____ TLNNNAME _____

CORRELATION
A      NAME      COLUMN NAME
*****
-----
----- = -----
----- = -----
----- = -----
----- = -----
----- = -----
----- = -----
----- = -----
----- = -----
----- = -----
```

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

You can also enter the INIT command to clear the join and restart the processing.

ACCESS COLUMNS

A field in which you can request a transfer to the Update SQL Join —Access Columns screen by entering any non-blank value.

TLNNAME

The CA Telon name the join used as the key name for the join. This name must be unique among join or table identifiers.

(Line command) A

A field where you can enter one of the following line commands:

C

Copy a line

CC

Copy a block of lines

I

Insert a line

Inn

Insert *nn* lines

IS(*n*)

Insert *n* lines for you to enter the command

M

Move a line

MM

Move a block of lines

R

Repeat a line

RR

Repeat a block of lines

A

Line(s) to insert, copy, or move go after this line

B

Line(s) to insert, copy, or move go before this line

D

Delete an entry

See Line Commands for more information.

CORRELATION NAME

The correlation name of the table used in the join selection criterion.

COLUMN NAME

The column name used in the join WHERE clause. The column name given must match a column name from the table with the correlation name and must be included in the join table identified by the TLNNAME field.

Update SQL Join—Access Columns

Access

On the Update SQL Join—Join Columns screen, enter any non-blank character in the ACCESS COLUMNS select field.

Program ID

D144

Function

Specifies the columns used in this join. The columns are qualified by the appropriate correlation name as specified on this screen.

UPDATE SQL JOIN - ACCESS COLUMNS **** QUAL.TBLNAME ***** SIZE 000000

COMMAND ==> _____ SCROLL ==> _____

JOIN COLUMNS ALIAS TLNNAME _____

CORRELATION

A NAME COLUMN NAME AC FROM QUAL.TABL

***** ***** ***** ** *****

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

JOIN COLUMNS

A field in which you can request to transfer to the Update SQL Join — Join Columns screen by entering any non-blank character.

ALIAS

Use this field to request display of the Add/Update SQL Join Alias screen where the ALIAS, KEY, and ACCESS information can be updated.

TLNNAME

The CA Telon name for the join, used as the key name for the join. This name must be unique among join or table identifiers.

(Line command) A

A field where you can enter one of the following line commands:

C

Copy a line

CC

Copy a block of lines

I

Insert a line

Inn

Insert *nn* lines

IS(*n*)

Insert *n* lines for you to enter the command

M

Move a line

MM

Move a block of lines

R

Repeat a line

RR

Repeat a block of lines

A

Line(s) to insert, copy, or move go after this line

B

Line(s) to insert, copy, or move go before this line

D

Delete an entry

See Line Commands for more information.

CORRELATION NAME

The correlation name of the table used in the join select criterion.

COLUMN NAME

The column name used in the join WHERE clause. The column name given must match the column name in the table with the correlation name.

AC

Displays the column's access value as identified on the Update SQL Join Alias screen.

FROM QUAL.TABLE

This field displays the fully-qualified table name for each column.

Update SQL Join—Alias

Access

On the Update SQL Join - Access Columns screen, enter any non-blank character in the ALIAS select field.

Program ID

D145

Function

Displays the columns used in this join and allows specification of the host variable name used in reading data from or inserting data into the join column.

```

***** ADD/UPDATE SQL JOIN ALIAS ** *****
COMMAND ==> _____ SCROLL ==> _____

          TLNNAME: _____ TABLE: _____

CORRELATION
  NAME      COLUMN NAME      KY/AC      ALIAS
***** *****
_____
  
```

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

TLNNAME

Displays the CA Telon name for the join.

TABLE

Displays the fully qualified join table name.

CORRELATION NAME

Displays the correlation name of the joined table containing the column.

COLUMN NAME

Displays the column name used in the join WHERE clause.

KY

A value to specify whether this column is involved in generated WHERE conditions. Values are:

Y

Declares this column a key column. CA Telon includes any such columns in sequential order in the generated WHERE condition.

P

Generates the WHERE condition to include this column in position p of the column order. If a number is specified for one column, Y is not a valid value for any column. The group of numbers must begin with 1 and be incremented by 1.

AC

A value to specify whether the EXEC SQL statement is to access the column. Values are:

Y

The column is accessed by the EXEC SQL statement

N

The column is not accessed by the EXEC SQL statement

ALIAS

Allows specification of a host variable name for the column used in reading data from or inserting data into the column.

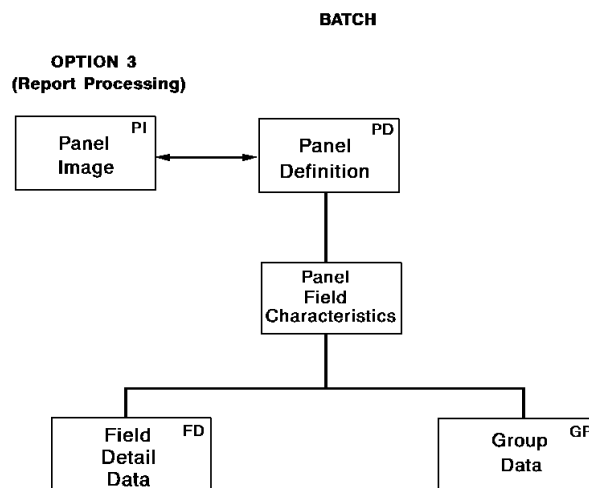
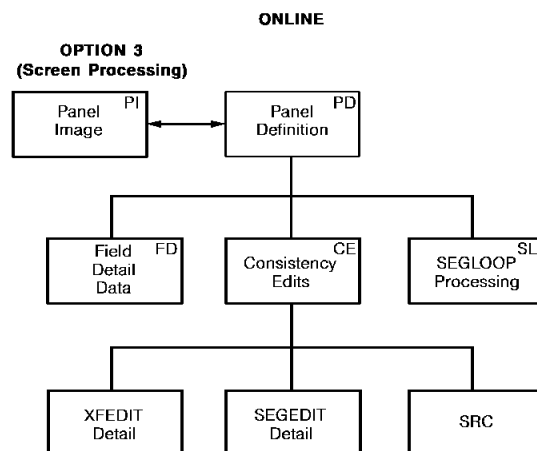
Note: CA-Datcom does not support the use of aliases. Any aliases entered for CA-Datcom tables are ignored.

Chapter 5: Panel Specification

Using panel definition, Option 3 on the TDF Main menu, you can define all the CA Telon statements that create portions of your program. The first steps in creating your program are to create:

- A panel image: the layout of a screen or report
- A panel definition: all the fields mapped to and from a program's panel image

This chapter documents the screens you associated with these steps. The following diagrams present the panel image and panel definition portion of a CA Telon-generated online program and batch program.



Panel image

You create a layout of a screen or report by keying the literal and variable fields in to the Edit Panel Image screen. CA Telon uses this image to capture field locations, field types, and field lengths.

Panel definition

The panel definition defines all CA Telon statements that create the portions of your program that:

- Control the terminal display
- Retrieve data from the terminal
- Store data from the terminal

The panel definition provides data such as:

- Data source being mapped to an output field
- Whether a particular field must be entered by the operator
- What edit tests a field must pass
- How data is reformatted during input
- What special attribute characters are used
- Where the data for the input field is mapped

The panel definition includes:

- Panel image
- Field characteristics
- CA Telon-generated consistency edits
- SEGLOOP processing

Panel Definition Menu

Access

On the TDF Main menu, enter **3**.

Program ID

P100

Function

Allows you to create and maintain panel images and panel definitions.

```

PANEL DEFINITION MENU *****
COMMAND ==> _____

FUNCTION:  __  CR-CREATE  UP-UPDATE  PU-PURGE  SH-SHOW  LI-LIST
ITEM      __  PI-IMAGE   PD-DEFIN
                        FD-FIELD   CE-CONSIG  SL-SEGL00P
                        (UP)       (CR,UP)   (CR,UP,PU)

MEMBER NAME:
HEADER  _____
ID      _____
DESC    _____

ENTER VALUE FOR SPECIFIC ITEM TO BE PROCESSED:
1. IMAGE      > < + \ (INPUT OUTPUT OUTIN SELECT LIT-BREAK CHARACTERS)
                24 080 (LINE-COLUMN IMAGE SIZE)
                U      (UPPER/LOWER CASE LITERALS)
2. DEFIN      Y Y Y Y N (INPUT OUTPUT OUTIN SELECT LITERAL FIELDS LISTED)
3. FIELD      _____ (NAME OR LINE,COLUMN OR "*PANEL")
4. CONSIG     _____ (TYPE - "XFEDIT", "SEGEDIT", OR BLANK FOR LIST)
                _____ (NAME - IF TYPE SPECIFIED)
5. SEGL00P    _____ (TYPE - "FILE" OR "TABLE")
                _____ (FROM NAME OR LINE,COLUMN)
                _____ (TO NAME OR LINE,COLUMN)

```

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

FUNCTION

Type of function to perform on the item identified by the values HEADER and ID fields.

The following table lists each function type and the items on which you can perform each function.

Value	Meaning	Items
CR	Create	PI PD CE SL
UP	Update	PI PD FD CE SL
PU	Purge	PI PD SL
SH	Show	PI PD
LI	List	PI PD

ITEM

The item to work with. Values are:

PI

Panel image

PD

Panel definition

FD

Field definition

CE

Consistency edit

SL

SEGLOOP

HEADER

A one- to five-character name that identifies a group of programs that form an application or portion of an application. This value must be unique for each group of programs.

The length of this field is determined at installation time. The combined length of HEADER and ID can be a maximum of six characters.

You can specify a HEADER value to correspond to a programming application group at your site. For example, CL could represent a claims application group.

ID

A one- to five-character name that identifies a particular program. For the program, the panel image, panel definition, screen definition, and report definition must have the same name.

For example, ABCD.PI, ABCD.PD, ABCD.SD, and ABCD.BD are all associated with the same program, whose ID is ABCD. The suffixes .PI, .PD, .SD, and .BD are appended by CA Telon for the panel image, panel definition, screen definition, and batch definition, respectively. See the following chapters later in this guide for more information about screen definitions and batch definitions.

- Online Program Definition
- Batch Program Definition

DESC

Description of the panel image or panel definition. For your informational purposes, CA Telon displays this description on screens such as the List Panel Definitions screen, the Edit Panel Image screen, and the Line Edit screen.

IMAGE

Characters that CA Telon uses to identify the field types on a panel image.

Values are any character except a single quote (') and an ampersand (&). However, each character used to define a field type may not be used in a literal field on a panel image, except for the literal break character. Default values are:

<

Input field

>

Output field

+

Outin field

|

Select field

\

Literal break

For your ID, you can change these CA Telon defaults on the Update Session Controls screen.

SIZE

Screen or report size, // X *ccc*, where:

//

Number of lines on the screen or report

ccc

Number of columns on the screen or report

The default is 24 x 80.

For your ID, you can change these CA Telon defaults on the Update Session Controls screen, subject to the following limitations:

Panel Characteristic	Maximum	Minimum
Panel columns	240	17
Panel lines	420 - Batch 0 99 - Online	1

Number of fields	2000	1
Number of groups	50	N/A
Screen size 0 (lines x columns)	9920	N/A

Note: If you are working with a Model 3 or Model 4 terminal, CA Telon displays the screen in Model 2 mode if you create a screen with 24 lines.

CASE

Value to specify whether to translate lower case literals into upper case literals. The value in this field is applicable only when the FUNCTION value is CR (CREATE). Values are:

U

Convert all literals to upper case

L

Save all literals as entered (mixed upper case and lower case)

The CASE field value affects only fields that are added or modified in the current session. You can modify the value when you are editing a panel with the CAPS command. See Primary Commands for more information.

DEFIN

Set of values to specify which field types are displayed on the Update Panel Fields (Online) screen. Values are:

Y

Display this field type

N

Do not display this field type

The order of values corresponds to the order of the field types as listed on the screen.

FIELD

Name, or the line and column position, of a field in a panel definition. A value in this field is not valid until the panel definition is created.

The value in this field has two possible functions:

- When the ITEM value is FD, it allows for direct field data detail update (bypassing the Update Panel Fields (Online) screen)
- When ITEM is CE, the value is one of the following:
 - Name of a select field for which the consistency edit is performed
 - *PANEL, for all consistency edits except those performed for select fields

CON SIS

Type of consistency edit. Valid when the FUNCTION field value is UP (update). Values, and the resulting screen displays, are:

XFEDIT

Update Cross-Field Edit (XFEDIT) screen

SEGEDIT

Update Segment Edit (SEGEDIT) screen

BLANK

List Consistency Edits screen

If a value is specified here, enter the consistency edit name in the NAME field.

NAME

Name of the consistency edit to edit directly from this screen.

If the edit was not named when it was created, you can access it only through the List Consistency Edits screen.

SEGLOOP

Type of mapping that SEGLOOP loop processing is to perform. Values are:

FILE

The program reads a segment or record for each iteration of the loop.

TABLE

The program reads data from or writes data to an array (table) for each iteration of the loop.

FROM

First variable field included in the SEGLOOP loop processing, identified by either its panel definition field name or by its line and column location.

TO

Last variable field included in the SEGLOOP loop processing, identified by either its panel definition field name or by its line and column location.

List Panel Definitions

Access

On the Panel Definition menu, enter **LI** in the FUNCTION field.

Program ID

P401

Function

Lists these items:

- Panel image (PI)
- Panel definition (PD)

Note: This screen lists a panel image only if there is no panel definition for the program. Entities displayed on the list vary depending on the screen from which the list is requested.

LIST DEFINITIONS *****				
COMMAND ==> _____		PAGE 01		
NAME	**RENAME**	*****DESCRIPTION*****	USER	UPDATE
XXXXXX.PD		PANEL DEFINITION FOR DOCUMENTATION	ABCDE	083085
YYYYYY.PD		PANEL DEFINITION FOR MODELING TEST	XYZ00	090385
ZZZZZZ.PI		PANEL IMAGE ONLY FOR MODELING TEST	XYZ00	091585
A				

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

(Function) A

A column position to the left of the NAME field in which you can enter a control character to manipulate the item on the line. Values are:

C

Copy

D

Change a description

P

Purge with confirmation

R

Rename

S

Show (browse)

U

Update

Z

Zap (purge without confirmation)

NAME

Name of the panel image or panel definition.

RENAME

One of several actions, depending upon the value in the FUNCTION field. When the action is complete, CA Telon returns a confirmation message.

Note: If the function field value for this entry is C or R, a value in the RENAME field is required.

Values, their meanings, and the associated messages are:

Value	Meaning	Message
C	Copy a member	*COPIED
D	Change the description of a member	*DESC UP
P	Purge a member with a	*PURGED

	confirmation	
R	Rename a member	*RENAMED
S	Enter a member in browse mode	*PNL SAVED
U	Enter a member in update mode	*PNL SAVED
Z	Purge a member without confirmation	*PURGED

CA Telon confirms a rename action, the old member no longer exists and information about it (that is, description, user, and update) is no longer displayed. Subsequent access to the data must be through the new name.

DESCRIPTION

Used for informational purposes. CA Telon requires this field for all functions, but it updates the DESCRIPTION field value only if the function field value is D.

USER

(Protected field.) Identifies the last user to access the member.

UPDATE

(Protected field.) Identifies the date of the last access of the member.

Edit Panel Image

Access

On the Panel Definition menu, enter:

- CR in the FUNCTION field
- PI in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field
- *Description* in the DESC field

Program ID

P103 (for screens)

P113 (for reports)

Function

Allows you to create a panel image, the first step in creating your program. The top line of the screen contains the description that you entered on the previous screen. You can modify the description by overwriting it here.

Press End to save the image. The message END PROCESSING PERFORMED appears on the Panel Definition submenu to indicate a successful save.

DESCRIPTION

Image characters

Image characters are the characters that CA Telon uses to identify the usage of a field painted on a panel image.

Values are any character except a single quote (') and an ampersand (&). However, the characters used to define field usage must be excluded from a literal field on a panel image, except for the literal break character.

Default image characters are:

<	Input field
>	Output field
+	Outin field
	Select field
\	Literal break

For example, if you accept the default character > for output fields, the TDF interprets > on a panel image as a position in an output field.

The CA Telon defaults can be overridden on the Update Session Controls screen. For more information, see Update Session Controls.

After you save a panel image using a set of image characters, you can change the image characters using the CH line command. For more information, see Line Commands.

Panel image example

This sample screen shows a panel image with literal, output, input, outin, and select fields. The variable fields are represented by the default image characters.

[illegible]

The output field shown on the last line of the screen is a field in which CA Telon can output error messages. This field:

- Can be located anywhere on the screen
- Can be 01 to 79 characters long
- We recommend that you make it at least 70 characters long
- Is required for all panel images used in screen definitions in online programs

Line Edit

Access

When creating a panel image, you can switch to line edit mode by pressing the PF key defined for it.

Program ID

P104 (for screens)

P114 (for reports)

Function

Allows you to move fields or areas of the image around, or to paint the image.

When you enter the line edit mode, CA Telon displays line command fields at the left of each line of the panel image, if it has been created.

To exit this screen, press End, SWAP EDIT, or LINE EDIT PF key.

LINE EDIT	MBTST1.PD *	PANEL 024 080	GROUP 001 OF 001	SIZE 000024	COL 002
COMMAND ==>	-----			SCROLL ==>	CSR
0001	A	DESCRIPTION			
0002					
0003					
0004					
0005					
0006					
0007					
0008					
0009					
0010					
0011					
0012					
0013					
0014					
0015					
0016					
0017					
0018					
0019					
0020					

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

(Line commands) A

A field that allows you to enter one of these line commands:

C

Copy a line

CC

Copy a block of lines

I

Insert a line

Inn

Insert *nn* lines

IS(*n*)

Insert *n* lines for you to enter the command

M

Move a line

MM

Move a block of lines.

R

Repeat a line

RR

Repeat a block of lines

A

Line(s) to insert, copy, or move go after this line

B

Line(s) to insert, copy, or move go before this line

D

Delete an entry

See Line Commands for more information.

Update Panel Fields (Online)

Access

On the Panel Definition menu, enter:

- **CR** or **UP** in the FUNCTION field
- **PD** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field

Note: The HEADER and ID values must match the corresponding values in the panel image, since both are part of the same program.

Program ID

P155

Function

Specifies the characteristics of all the fields in the panel image, the first step in creating a panel definition.

```
hhiiii.PD UPDATE PANEL FIELDS *****
COMMAND ==> PAGE 01
OPTIONS ==> ATTRS _ HELPMMSG _ MAPOUT _
LINE 001 COL 001 SIZE 024 080
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----
0001 DESCRIPTION
0002
0003
-----
U LN COL LTH USE **NAME** *FLDTYPE* ***** DBNAME OR TEXT ***** REQ MORE
  01 029 011 LI DESCRIPTION
  04 002 001 IN IN1 FILE-FIELD1 Y +

A
```

Field Definitions

COMMAND

For information, see Primary Commands.

OPTIONS

For a screen definition, a set of one-byte fields that allow you to transfer to a specified screen. Enter a nonblank character in the field that represents the screen of your choice:

ATTRS

Update Attribute Parameter screen

HELPMMSG

Update Help Message Parameters screen

MAPOUT

Update Mapout Parameter screen

LINE COL

The panel image lines to view on the three-line panel image display window. Change the window by altering the LINE and COL values and pressing Enter. The line and column number image location is displayed in the upper left corner of the window.

Panel image display window

A display-only three-line window to the panel image. You can change the image's field position, content, or type by changing values in the remaining fields on this screen.

U

The action to take on a field in the panel image. Values are:

U

CA Telon displays the appropriate screen:

- Update Literal Fields
- Update Output/Input/Outin Field
- Update Select Fields

D

Delete the field

Z

Delete the field

I

Insert a blank line after the field in order to enter another field

LN COL LTH USE

The starting line number, starting column number, length, and use respectively of the field. Possible USE values are:

LI

Literal field

IN

Input field

OU

Output field

OI

Outin field

SE

Select field

You can change these characteristics by entering new values. Also, you can add a new field to the panel image by specifying these values immediately below the last field on the screen. Once you enter the new data, CA Telon moves it to its relative position in the existing field information, and you can update it.

NAME

The field name. CA Telon uses this value to generate MFS or BMS control blocks and create COBOL or PL/I buffer names. CA Telon gives these names a prefix of TPI- for input buffer names and TPO- for output buffer names.

The name must be one to 8 characters long. The first character must be a letter, @, #, or \$. Subsequent characters may be numbers. No other special characters are allowed. Otherwise, Generator errors will result.

You refer to this value in a consistency edit, segment edit, or auto cursor position field.

Literal fields require a NAME value only when referenced from the program. When a name is specified for a literal field, it is included in the BMS map for CICS or MFS for IMS. A NAME value is required for all other fields. If not specified, CA Telon supplies a default name P//ccc, where // and ccc are starting line and column positions. For example, P20005 is for a field beginning on line 20 in column 5.

Field names ERRMSG1, MORE, PAGENO, SYSMSG, TRANCDE, TRANFLD, or PASSWORD have special meaning to CA Telon. If used, the FLDTYPE value is normally set to NONE.

FLDTYPE

The mapping of data to or from the DBNAME, with or without editing. Values are:

ALPHA

Direct alphanumeric move.

NONE

No automatic move. Programmers must perform any mapping required.

NUMERIC

Check for a numeric value or CA Telon specific field edits (for example: DATE, DOLLAR, and STATE).

field-edit

An application-specific field edit name. A call is made to *field-edit*, prefixed with I or O for input or output (where a subroutine must be created to perform this edit). See Field Edit Formats for more information.

Default values are:

- ALPHA, if a DBNAME is specified but no RANGE is supplied
- NUMERIC, if a DBNAME and a RANGE are supplied
- NONE, if no DBNAME is supplied

This specification is valid for output, input, outin, and select usage types. For outin, two edit names are necessary if output and input processing are different.

If the field edit fails, control returns to the program without processing.

Editing occurs in:

- E-100-INPUT-EDITS for input fields
- B-100-OUTPUT-EDITS for output fields
- E-100-INPUT-EDITS and B-100-OUTPUT-EDITS for outin fields
- J-100-SELECT for select fields

Note: A FLDTYPE value may not be specified if a CONVERT, VALUES, or FORMAT specification is supplied. For more information, see Update Output/Input/Outin Field.

DBNAME OR TEXT

The mapping name. On an output field, it indicates the program data field that is mapped out to the screen display. On an input field, it indicates the data field to receive the data from the screen.

An input field can be mapped to two different program fields, and an outin field can be mapped out from one data field and mapped back into two different fields. To designate these additional fields, you use the appropriate screen to update field data:

- Update Literal Fields
- Update Output/Input/Outin Field
- Update Select Fields

With PL/I programs, you can use any expression that is valid in an assignment or external procedure call. If the name is qualified by a structure name, include that structure name before the data name in this format *structure-name.data-name*. COBOL programs accomplish this function with the value in the OF field of the Update Output/Input/Outin Field screen.

If you enter a SHOW command in the COMMAND field, the command is displayed in this field.

Note: CA Telon does not allow you to overwrite literals with spaces on this screen. If you attempt to overwrite a literal, CA Telon places an asterisk (*) in the field and displays a message that text is required. If you press Enter or PF3, * becomes the default literal.

REQ

A value to specify whether the application user must enter data in an input or outin field. Values are:

Y

The application user must enter data.

N

(Default.) The application user does not have to enter data. If the user leaves the field blank, its value is set to zeros if field type is NUMERIC or spaces if field type is ALPHA.

C

Consistency edit code checks to see if this field is required. If a field with type C contains the required asterisk (*) character, the field is treated as if it were blank.

MORE

A field that signifies, by displaying a plus sign (+), that there is additional detail data for this field that cannot be displayed on this screen. To display the additional data (or enter additional data), enter **U** in the U field; this transfers you to the appropriate screen:

- Update Literal Fields
- Update Output/Input/Outin Field
- Update Select Fields

SEGLOOP DELIMITERS A

(*Protected field.*) Displays information about the range of the SEGLOOP:

- Under the NAME heading, the literal "SEGLOOP"
- Under the FLDTYPE heading, the SEGLOOP type (FILE or TABLE)
- Under the DBNAME OR TEXT heading, the SEGLOOP attribute (that is, OUTPUT, OUTIN, or INPUT)

The following information is displayed:

- The literal "SEGLOOP -"
- The literal "START" or "END" to indicate that this is the beginning or ending delimiter of the SEGLOOP

The line that is immediately after the START SEGLOOP delimiter becomes the first field in the SEGLOOP. The line that is immediately before the LAST SEGLOOP delimiter becomes the last field in the SEGLOOP.

Valid line commands for the SEGLOOP delimiters are:

U

Transfers you to the appropriate SEGLOOP detail screen:

- Create/Update Table SEGLOOP
- Create/Update File SEGLOOP

Z

Deletes the SEGLOOP

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

LINE COL

The panel image lines to view on the three-line panel image display window. Change the window by altering the LINE and COL values and pressing Enter. The line and column number image location is displayed in the upper-left corner of the window.

(Panel image display window) A

A display-only three-line window to the panel image. You can change the image's field position, content, or type by changing values in the remaining fields on this screen.

LN COL LTH USE

The starting line number, starting column number, length, and use respectively of the field. Possible USE values are:

LI

Literal field

IN

Input field

OU

Output field

OI

Outin field

SE

Select field

NAME

The field name, from the Update Panel Fields (Online) screen. See Update Panel Fields (Online) for more information.

HELPMMSG

Specify the name of the key used by the help facility to bring up the appropriate screen.

Note: When you enter a value here, you also enter (by default) a **Y** in the HELP field of the Update/Show Screen Parameters screen. See Update/Show Screen Parameters, for more information.

Update Mapout Parameter

Access

On the Update Panel Fields (Online) screen, enter any nonblank character in the MAPOUT field.

Program ID

P159

Function

Performs batch updates of the MAPOUT value.

XXXXXX.PD UPDATE MAPOUT PARM *****

COMMAND ==> PAGE 01

LINE 001 COL 001 SIZE 024 080

-----+---1---+---2---+---3---+---4---+---5---+---6---+---7---

0001 DESCRIPTION

0002 A

0003

LN COL LTH USE **NAME** ***** MAPOUT *****

Field Definitions

COMMAND

For information, see Primary Commands.

LINE COL

The panel image lines to view on the three-line panel image display window. Change the window by altering the LINE and COL values and pressing Enter. The line and column number image location is displayed in the upper-left corner of the window.

(Panel image display window) A

A display-only three-line window to the panel image. You can change the image's field position, content, or type by changing values in the remaining fields on this screen.

LN COL LTH USE

The starting line number, starting column number, length, and use respectively of the field. Possible USE values are:

LI

Literal field

IN

Input field

OU

Output field

OI

Outin field

SE

Select field

NAME

The field name, from the Update Panel Fields (Online) screen.

See Update Panel Fields (Online) for more information.

MAPOUT

The data name of any COBOL or PL/I variable in the program that controls output mapping of the output or outin field identified on this line. When the value of the variable is Y, the associated field is mapped to the output buffer by code that CA Telon generates in the B-100-OUTPUT-EDITS section. If the value is not Y, the field is not mapped.

The field that is used for MAPOUT must be set in custom code, and the MAPOUT field must be specified on each detail screen for the fields that are conditionally mapped on output. When CA Telon generates the code with the MAPOUT field coded, it sets up an IF statement before each individual output field.

A list of all MAPOUT fields for the entire screen also appears on the Update Output/Input/Outin Field screen.

Update Attribute Parameter

Access

On the Update Panel Fields (Online) screen, enter any nonblank character to select the ATTR option.

Program ID

P158

Function

Performs batch updates of field attributes.

```
XXXXXX.PD UPDATE ATTRIBUTE PARMS***** *****
COMMAND ==>                                     PAGE 01
LINE 001 COL 001                                SIZE 024 080
-----+---1---+---2---+---3---+---4---+---5---+---6---+---7---
0001                                DESCRIPTION
0002                                A
0003
-----
  LN COL LTH USE **NAME**  ATTRPRO ATTRINT  OUTATTR  EACOLOR  EAHIGH  EVALID
  01 029 011 LI
  04 002 001 IN  IN1      -    - - - -    -    - -    - -    -
```

Field Definitions

COMMAND

For information, see Primary Commands.

LINE COL

The panel image lines to view on the three-line panel image display window. Change the window by altering the LINE and COL values and pressing Enter. The line and column number image location is displayed in the upper left corner of the window.

(Panel image display window) A

A display-only three-line window to the panel image. You can change the image's field position, content, or type by changing values in the remaining fields on this screen.

LN COL LTH USE

The starting line number, starting column number, length, and use respectively of the field. Possible USE values are:

LI

Literal field

IN

Input field

OU

Output field

OI

Outin field

SE

Select field

NAME

The field name, from the Update Panel Fields (Online) screen. See Update Panel Fields (Online) for more information.

ATTRPRO

Specifies that a field is protected against application user entry when the field is written to the terminal. It is valid on input, outin, and select fields. Values are:

Y

Protect the field

N

Do not protect the field

Attribute protection is most often used on output fields to prevent the application user from changing a displayed value that is returned to the program on input.

Note: If the ATTRPRO value is Y, you can unprotect the field by moving the reserved attribute fields (for example, OK-ATTR or CURSOR-ATTR) to it. See the *Programming Concepts Guide* for information on these fields.

ATTRINT

The intensity of the field displayed. Values are:

NORMAL

(Default.) Normal intensity.

HIGH

High intensity

(blank)

Do not display the field

Note: You can change the field intensity at a later time by moving the reserved attribute fields (for example, INPUT-BLANK-ATTR or INPUT-HIGH-ATTR) to it. See the *Programming Concepts Guide* for information on reserved attribute fields.

OUTATTR

A value to specify whether the attributes for this field are included in the CA Telon screen image area located at the end of the Transfer Work Area of all CA Telon-generated programs. This causes the screen image to be larger, but this is required when you use CA Telon Help or Hold or if you must refresh the screen. Values are:

Y

(Distributed default.) Include attributes in the output buffer

N

Do not include attributes in the output buffer

This value overrides the installation default.

EACOLOR

The color displayed on an extended attribute terminal. This value is valid only when the EATTR value on the Update/Show Screen Parameters screen is Y. Values are:

BL

Blue

RE

Red

PI

Pink

TU

Turquoise

YE

Yellow

GR

Green

DE

The value set at screen definition

NE

Neutral (usually white, depending on installation default)

EAHIGH

The extended highlight attribute for the field. This value is valid only when the EATTR value on the Update/Show Screen Parameters screen is Y. Values are:

B

Blink when displayed

BL

Blink when displayed

R

Display in reverse video

RE

Display in reverse video

D

Display in default mode

DE

Display in default mode

U

Underline display

UN

Underline display

EVALID

The extended validation attribute for the field. This value is valid only when the EATTR value on the Update/Show Screen Parameters screen is Y and you request that CA Telon generate one of the following types of source code:

- IMS MFS, on the Update TSO or IMS Screen Environment screen
- CICS BMS, on the Update CICS Screen Environment screen

Values are:

MF

The application user must enter all character positions of the field

ME

The application user must enter at least one character into the field

BO

Both of the above apply

During execution, terminal input in the field is checked before control is returned to the program. Control is returned to the program only if something is entered. By comparison, the REQ field value is used to check for input from inside the program.

See Update Panel Fields (Online) for more information on the REQ field.

If the application uses terminals both with and without extended attributes and the EVALID is ME, the REQ value must be Y.

Update Literal Fields

Access

On the Update Panel Fields (Online) screen, enter **U** in the U field.

Program ID

P180

Function

Adds or updates details for a literal field.

```
XXXXXX.PD UPDATE LITERAL FIELD ***** *****  
COMMAND ==> _____  
  
FIELD NAME _____ USAGE LITERAL LINE 01 COL029 LTH 011  
MAPPING: TEXT _____  
* _____  
* _____  
* _____  
GENERAL: OUTATTR (Y/N)  
ATTR: ATTRINT _____ EACOLOR __ EAHIGH __
```

Field Definitions

COMMAND

For information, see Primary Commands.

FIELD NAME

The field name, from the Panel Definition menu or the Update Panel Fields (Online) screen. See Panel Definition Menu or [:hdref refid=p155](#). for more information.

USAGE

The type of field. Values are:

- LITERAL (*Default*)
- SELECT
- INPUT
- OUTPUT
- OUTIN

A value is required if a value is not carried over from the Panel Definition menu or the Update Panel Fields (Online) screen.

LINE COL LTH

(*Protected field.*) Displays the starting line number, starting column number, and length respectively of the literal field. These values were specified on the Panel Definition menu or the Update Panel Fields (Online) screen.

TEXT

The text for a literal field. Do not use the single quote (') or an ampersand (&). characters.

OUTATTR

Specifies whether the attributes for this field are included in the CA Telon screen image area located at the end of the Transfer Work Area of all CA Telon-generated programs. This causes the screen image to be larger, but this is required when you use CA Telon Help or Hold or if you must refresh the screen. Values are:

Y

(*Distributed default.*) Include attributes in the output buffer

N

Do not include attributes in the output buffer

This value overrides the installation default.

ATTRINT

The intensity of the field displayed. Values are:

NORMAL

(Default.) Normal intensity.

HIGH

High intensity

(blank)

Do not display the field

Note: You can change the field intensity at a later time by moving the reserved attribute fields (for example, INPUT-BLANK-ATTR or INPUT-HIGH-ATTR) to it. See the *Programming Concepts Guide* for information on reserved attribute fields.

EACOLOR

The color displayed on an extended attribute terminal. This value is valid only when the EATTR value on the Update/Show Screen Parameters screen is Y. Values are:

BL

Blue

RE

Red

PI

Pink

TU

Turquoise

YE

Yellow

GR

Green

DE

The value set at screen definition

NE

Neutral (usually white, depending on installation default)

EAHIGH

The extended highlight attribute for the field. This value is valid only when the EATTR value on the Update/Show Screen Parameters screen is Y. Values are:

B

Blink when displayed

BL

Blink when displayed

R

Display in reverse video

RE

Display in reverse video

D

Display in default mode

DE

Display in default mode

U

Underline display

UN

Underline display

Update Output/Input/Outin Field

Access

On the Update Panel Fields (Online) screen, enter **U** in the U column for a listed field with a USE value of OU (output), IN (input), or OI (outin).

Note: The title displayed on this screen varies to reflect the usage of the field being updated.

Program ID

P181

Function

Specifies additional data for a given field.

```

XXXXXX.PD UPDATE OUTIN  FIELD ***** *****
COMMAND ==>
  FIELD NAME HHH_____ USAGE OUTPUT_  LINE 02 COL 002 LTH 005

GENERAL IN: REQ (Y/N/C)                      HELPMMSG _____
*      OUT: PIC _____ OUTATTR (Y/N) _____

MAPPING: DBNAME _____
*      OF _ A _____
*      .....
*      OF _____
*      .....
*      OF _____
*      INIT _____
*      MAPOUT _____

EDIT:  FLDTYPE OUT _____ IN _____ PARM LIST EXTENSION
*      SPEC _____ (FORMAT/CONVERT/VALUES/RANGE)
*
*
*
*

ATTR:  ATTRPRO ATTRINT _____ EACOLOR EAHIGH EVALID
      FMTEXTIT _____ FMTCNTL=MFS (Y/N)

```

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

FIELD NAME

The field name, from the Panel Definition menu or the Update Panel Fields (Online) screen.

See Panel Definition Menu or Update Panel Fields (Online) for more information.

USAGE

The type of field. Values are:

- LITERAL. (*Default*)
- SELECT
- INPUT
- OUTPUT
- OUTIN

A value is required if no value is carried over from the Panel Definition menu or the Update Panel Fields (Online) screen.

LINE COL LTH

(*Protected field.*) Displays the starting line number, starting column number, and length respectively of the literal field. These values were specified on the Panel Definition menu or the Update Panel Fields (Online) screen.

REQ

A value to specify whether the application user must enter data in an input or outin field. Values are:

Y

The application user must enter data.

N

(*Default.*) The application user does not have to enter data. If the user leaves the field blank, its value is set to zeros if the field type is NUMERIC or spaces if field type is ALPHA.

C

Consistency edit code checks to see if this field is required. If a field with type C contains the required asterisk (*) character, the field is treated as if it is blank.

HELPMSG

The field-level help message, usually a key used by the help facility to bring up the appropriate screen.

Note: When you enter a value here, you also enter **Y** in the **HELP** field of the Update/Show Screen Parameters screen. For more information, see Update/Show Screen Parameters.

PIC

The COBOL or PL/I picture clause (any valid COBOL or PL/I numeric format). This value overrides the standard output formatting by CA Telon based on the **FLDTYPE** value, except that a **FLDTYPE** value of **FLOAT** overrides the **PIC** value (for example, **\$\$9.9**).

The standard output formatting for a five-digit number based on **FLDTYPE** value is:

FLDTYPE Value	COBOL	PL/I
NUMERIC	Z(4)9	(4)Z9
FULLNUM	9(5)	(5)9
DOLLAR	Z(2)9.99	(2)Z9.99<%rrule 2 3%>

OUTATTR

A value to specify whether the attributes for this field are included in the output buffer. This field is valid for only output fields and literal fields with field names. Values are:

Y

Include attributes in the output buffer

N

Do not include attributes in the output buffer

This value overrides the installation default.

DBNAME

The file, working storage, or transfer work area field name that CA Telon uses to map data to and/or from the field represented on the screen.

Note: To have CA Telon generate special code to support the COBOL II "ACCEPT ... FROM DATE" or COBOL/LE "MOVE FUNCTION CURRENT-DATE ..." use the reserved word **@DATEV**. This reserved word is appropriate for OUTPUT only; it should not be used for INPUT-only fields. A separate INPUT **DBNAME** must be specified for OUTIN fields.

OF A

(COBOL Only.) A qualification of the DBNAME field value (for example, TAX-DUE OF FEDERAL-RETURN). For USAGE values INPUT, OUTIN, and SELECT, you can map to and/or from more than one field (DBNAME).

11. INIT

The initialized value of an input or select field when written to the terminal. Omit single quotes (') or ampersands (&). from this value. Otherwise, errors will occur during program assembly.

12. MAPOUT

The data name of any COBOL or PL/I parameter in the program that controls output mapping of the field represented on this line. See Update Mapout Parameter for more information.

FLDTYPE

The mapping of data to or from the DBNAME, with or without editing. Values are:

ALPHA

Direct alphanumeric move.

NONE

No automatic move. Programmers must perform any mapping required.

NUMERIC

Check for a numeric value or CA Telon specific field edits (for example: DATE, DOLLAR, and STATE).

field-edit

An application-specific field edit name. A call is made to *field-edit*, prefixed with I or O for input or output (where a subroutine must be created to perform this edit). See :hhref page=no refid=AA00000. for more information.

Default values are:

- ALPHA, if a DBNAME is specified but no RANGE is supplied
- NUMERIC, if a DBNAME and a RANGE are supplied
- NONE, if no DBNAME is supplied

This specification is valid for output, input, outin, and select usage types. For outin, two edit names are necessary if output and input processing are different.

If the field edit fails, control returns to the program without processing.

Editing occurs in:

- E-100-INPUT-EDITS for input fields
- B-100-OUTPUT-EDITS for output fields
- E-100-INPUT-EDITS and B-100-OUTPUT-EDITS for outin fields
- J-100-SELECT for select fields

Note: A FLDTYPE value may not be specified if the SPEC value is CONVERT, VALUES, or FORMAT.

PARM LIST EXTENSION

A field to request transfer to the Update Parameter List Extension screen by entering any nonblank character. Use this screen to specify extended fields for installation-defined FLDTYPE edits.

To transfer, a value must already have been entered in the FLDTYPE field.

SPEC

The edit specification:

FORMAT

Defines a mask used to format an alphanumeric field on input or output.

CONVERT

Defines acceptable display values and specifies how the program stores them. It can be used for input or output mapping.

RANGE

Defines one or more acceptable ranges of values for a numeric field on input.

VALUES

Defines one or more valid input values for input, outin, and select fields.

Enter the specification in the field and its associated values in the extended field space below the SPEC field. The syntax for entering each specification and a full description of its function follow.

FORMAT *mask*

In the **FORMAT** *mask* 9s represent numbers and Xs represent characters. All other characters are inserted into the corresponding positions on output and stripped from the corresponding positions on input. Here are examples of how an input value is stored after formatting by the mask and how that same value is displayed on output through the same mask:

Input	Mask	Stored value	Output
123-45-678	999-99-999	12345678	123-45-678
ABCD343	ABXX993	CD34	ABCD343
RTXY887	ABXX993	XY88	ABXY883
CDQ34	XXT99	CD34	CDT34

All characters except the mask-defined value 9 are stripped for storage. CA Telon does not verify the value in X position on input. (In the third example, CA Telon accepts but does not store the input value RT.) CA Telon does verify that characters specified as 9s are numbers.

If the length of the *mask* does not equal the length of the input field, the target is blank-padded to the right during mapping. Remember, the target data field must be alphanumeric even if the mask is all 9s.

CONVERT *screen-val-1,stored-val-1 [,screen-val-2,stored-val-2 ...]*

Screen-val is the value as it appears on the screen; *stored-val* is the value as it is stored. If you define more than one pair, the length of all occurrences of *screen-val* must be the same; this is also true for all occurrences of *stored-val*. If you use blanks to pad the length, enclose the value in single quotes.

In this example, the two pairs specified result in the indicated screen value and stored value:

```
CONVERT FEMALE,F,'MALE ',M
Valid screen value    Stored value
FEMALE                F
MALE                  M
```

For input or select fields, CA Telon puts the pairs in a table for searching at runtime. During execution, the application user's input must match a specified *screen-val*. If not, CA Telon flags the field and an error is returned.

If there is no corresponding *stored-val* during output, the program displays the stored value as is.

RANGE *start-range-1,end-range-1* [,*start-range-2,end-range-2* ...]

Start-range and *end-range* can be numeric constants or data field names. Values must be specified lowest to highest. The range is inclusive; that is, the numbers defining the range are themselves within the range.

In the following example, valid input values for the field are defined as between 4 and 9, 23 and 33, or 53.8 and 75.

```
RANGE 4, 9, 23, 33, 53.8, 75
```

VALUES *value-1* [,*value-2* ...]

Using the specified values, CA Telon generates COBOL 88-level items or a PL/I search array. At runtime, these are the only acceptable values. All others are flagged as errors. In the following example, only the values 10, 12, and AL are acceptable.

```
VALUES 10, 12, AL
```

Although not required, all values should be the same length as the LTH value for the field. If the entered value is longer, at compile time COBOL issues a diagnostic and PL/I truncates the length to the LTH value.

ATTRPRO

A value to specify that a field is protected against application user entry when the field is written to the terminal. It is valid on input, outin, and select fields. Values are:

Y

Protect the field

N

Do not protect the field

Attribute protection is most often used on outin fields to prevent the application user from changing a displayed value that is returned to the program on input.

Note: If the ATTRPRO value is Y, you can unprotect the field by moving the reserved attribute fields (for example, OK-ATTR or CURSOR-ATTR) to it. See the *Programming Concepts Guide* for information on these fields.

ATTRINT

The intensity of the field displayed. Values are:

NORMAL

(Default.) Normal intensity.

HIGH

High intensity

(blank)

Do not display the field

Note: You can change the field intensity at a later time by moving the reserved attribute fields (for example, INPUT-BLANK-ATTR or INPUT-HIGH-ATTR) to it. See the *Programming Concepts Guide* for information on reserved attribute fields.

EACOLOR

The color displayed on an extended attribute terminal. This value is valid only when the EATTR value on the Update/Show Screen Parameters screen is Y. Values are:

BL

Blue

RE

Red

PI

Pink

TU

Turquoise

YE

Yellow

GR

Green

DE

The value set at screen definition

NE

Neutral (usually white, depending on installation default)

EAHIGH

The extended highlight attribute for the field. This value is valid only when the EATTR value on the Update/Show Screen Parameters screen is Y. Values are:

B

Blink when displayed

BL

Blink when displayed

R

Display in reverse video

RE

Display in reverse video

D

Display in default mode

DE

Display in default mode

U

Underline display

UN

Underline display

EVALID

The extended validation attribute for the field. This value is valid only when the EATTR value on the Update/Show Screen Parameters screen is Y and you request that CA Telon generate one of the following types of source code:

- IMS MFS, on the Update TSO/IMS Screen
- CICS BMS, on the Update CICS Screen Environment screen

Values are:

MF

The application user must enter all character positions of the field

ME

The application user must enter at least one character into the field

BO

Both of the above apply

During execution, terminal input in the field is checked before control is returned to the program. Control is returned to the program only if something is entered. By comparison, the REQ field value is used to check for input from inside the program. See Update Panel Fields (Online) for more information on the REQ field.

If the application uses terminals both with and without extended attributes and the EVALID is ME, the REQ value must be Y.

FMTEXT

(IMS applications only.) An MFS field exit routine invoked for the field on input. Two values are required for this specification:

- Exit Routine Number (values: 000 through 127)
- Exit Routine Value (values: 000 through 255)

FMTCTL

(IMS applications only.) A value to specify that this is an output field displaying special MFS data.

Values are:

Y

This is a special MFS field

N

This is not a special MFS field

If FMTCTL is Y, the FIELD NAME value (LABEL field value on other screens) must be one of the MFS system literals:

- DATE1
- DATE2
- DATE3
- DATE4
- TIME
- PASSWORD
- LTSEQ
- LTNAME
- LPAGENO
- LTMSG

For fields with labels other than PASSWORD, the field usage is assumed to be OUTPUT (other usage specifications are ignored). The field occupies space on the screen, but does not appear in the buffers (that is, the CA Telon program knows nothing about the field).

Update Select Fields

Access

Access this screen in one of these ways:

- On the Update Panel Fields (Online) enter **U** in the U field for a select field
- On the Update Select Parameters screen, enter a nonblank value to select the SELECT FIELDS option

Program ID

P182

Function

Add or update information about a given select field.

```

XXXXXX.PD UPDATE SELECT FIELD *****
COMMAND ==>
FIELD NAME AA _____ USAGE SELECT LINE 03 COL 003 LTH 041

GENERAL: NEXTPGM _____ SCONSIS _____ HELPMMSG _____
*   INEDIT (Y/N) INDBIO (Y/N)
*   SELKEY FROM _____
*           TO _____

MAPPING: DBNAME1 _____
*         OF _____
*   DBNAME2 _____
*         OF _____
*   INIT _____

EDIT:  FLDTYPE _____ PARM LIST EXTENSION
*     SPEC _____ (FORMAT/CONVERT/VALUES/RANGE)
*
*
*
*

ATTR: ATTRPRO _____ ATTRINT _____ EACOLOR _____ EAHIGH _____ EAVALID _____
      FMTEXTIT _____ FMTCNTL=MFS (Y/N)

```

Field Definitions

COMMAND

For information, see Primary Commands.

FIELD NAME

The field name, from the Panel Definition menu or the Update Panel Fields (Online) screen. See Update Panel Fields (Online) and Panel Definition Menu for more information.

USAGE

The type of field (SELECT).

LINE COL LTH

(Protected field.) Displays the starting line number, starting column number, and length respectively of the literal field. These values were specified on the Panel Definition menu or the Update Panel Fields (Online) screen.

NEXTPGM

The ID of the next program in the application to receive control from the current program. The length of this field is determined at CA Telon installation.

If a value is not specified here, program control can be handled using PF-key or consistency-edit coding.

Note: This value is overridden by the value, if any, specified in the NEXTPGM field of Create/Update Screen Definition screen.

SCONSIS

For input processing, the member name of the statement copied or included in the J-100 paragraph execution for a select field.

HELPMSG

The field-level help message, usually a key used by the help facility to bring up the appropriate screen.

Note: When you enter a value here, you also enter **Y** in the HELP field of the Update/Show Screen Parameters screen. See Update/Show Screen Parameters for more information.

INEDIT

Specifies whether CA Telon is to execute the E-100-INPUT-EDITS section before NEXTPGM and/or SCONSIS processing. CA Telon generates the E-100 automatically if the program contains select fields. The E-100-INPUT-EDITS section edits input fields according to FLDTYPE specification. Values are:

Y

CA Telon executes the E-100-INPUT-EDITS section automatically. If errors occur, CA Telon skips the balance of processing and returns an error message to the screen.

N

(Default.) CA Telon does not automatically execute the E-100-INPUT-EDITS section; however, you can explicitly call it from SCONSIS code.

INDBIO

A value to specify whether CA Telon is to perform the H-100 section to create and update segments requested for auto exec. CA Telon generates the H-100 paragraph if the program contains select fields. Values are:

Y

CA Telon automatically executes the H-100-INPUT-TERM section

N

(Default.) CA Telon does not automatically execute the H-100-INPUT-TERM section; however, you can explicitly call it from SCONSIS code

SELKEY

The source and target fields for the key value of a line displayed on the screen for further processing. This specification is meaningful when input to the select field is an identifier of an item displayed within list processing.

The two subfields that represent the range of key values used in a SEGLOOP entry are:

FROM

The name of the data item in which key information was stored during output processing (generally defined using the SAVEKEY parameter of the SEGLOOP statement)

TO

The name of the data item in which the key is stored for later processing

DBNAME

The name of the file, working storage, or transfer work area that CA Telon uses to map data to and/or from the select field.

OF (COBOL only.)

A qualification of the DBNAME field value (for example, TAX-DUE OF FEDERAL-RETURN). For USAGE values INPUT, OUTIN, and SELECT, you can map to and/or from more than one field (DBNAME).

INIT

The initialized value of an input or select field when written to the terminal. Omit single quotes (') or ampersands (&). from this value. Otherwise, errors will occur during program assembly.

FLDTYPE

The mapping of data to or from the DBNAME, with or without editing. Values are:

ALPHA

Direct alphanumeric move.

NONE

No automatic move. Program must perform any mapping required.

NUMERIC

Check for a numeric value or CA Telon specific field edits (for example: DATE, DOLLAR, and STATE).

field-edit

An application-specific field edit name. A call is made to *field-edit*, prefixed with I or O for input or output (where a subroutine must be created to perform this edit).

See Appendix A, Field Edit Formats for more information.

Default values are:

- ALPHA, if a DBNAME is specified but no RANGE is supplied
- NUMERIC, if a DBNAME and a RANGE are supplied
- NONE, if no DBNAME is supplied

This specification is valid for output, input, outin, and select usage types. For outin, two edit names are necessary if output and input processing are different.

If the field edit fails, control returns to the program without processing.

Editing occurs in:

- E-100-INPUT-EDITS for input fields
- B-100-OUTPUT-EDITS for output fields
- E-100-INPUT-EDITS and B-100-OUTPUT-EDITS for outin fields
- J-100-SELECT for select fields

Note: A FLDTYPE value may not be specified if the SPEC value is CONVERT, VALUES, or FORMAT.

PARM LIST EXTENSION

A field to request transfer to the Update Parameter List Extension screen by entering any nonblank character. Use this screen to specify extended fields for installation-defined FLDTYPE edits.

To transfer, a value must already have been entered in the FLDTYPE field.

SPEC

The edit specification:

FORMAT

Defines a mask used to format an alphanumeric field on input or output.

CONVERT

Defines acceptable display values and specifies how the program stores them. It can be used for input or output mapping.

RANGE

Defines one or more acceptable ranges of values for a numeric field or input.

VALUES

Defines one or more valid input values for input, outin, and select fields.

Enter the specification in the field and its associated values in the extended field space below the SPEC field. See the discussion of the SPEC field in Update Output/Input/Outin Field for details about the syntax for entering each specification and a full description of its function.

ATTRPRO

Specifies whether a field is protected against application user entry when the field is written to the terminal. It is valid on input, outin, and select fields. Values are:

Y

Protect the field

N

Do not protect the field

Attribute protection is most often used on outin fields to prevent the application user from changing a displayed value that is returned to the program on input.

Note: If the ATTRPRO value is Y, you can unprotect the field by moving the reserved attribute fields (for example, OK-ATTR or CURSOR-ATTR) to it. For information on these fields, see the *Programming Concepts Guide*.

ATTRINT

The intensity of the field displayed. Values are:

NORMAL

(Default.) Normal intensity

HIGH

High intensity

(blank)

Do not display the field

Note: You can change the field intensity at a later time by moving the reserved attribute fields (for example, INPUT-BLANK-ATTR or INPUT-HIGH-ATTR) to it. For information on reserved attribute fields, see the *Programming Concepts Guide*.

EACOLOR

The color displayed on an extended attribute terminal. This value is valid only when the EATTR value on the Update/Show Screen Parameters screen is Y. Values are:

BL

Blue

RE

Red

PI

Pink

TU

Turquoise

YE

Yellow

GR

Green

DE

The value set at screen definition

NE

Neutral (usually white, depending on installation default)

EAHIGH

The extended highlight attribute for the field. This value is valid only when the EATTR value on the Update/Show Screen Parameters screen is Y. Values are:

B

Blink when displayed

BL

Blink when displayed

R

Display in reverse video

RE

Display in reverse video

D

Display in default mode

DE

Display in default mode

U

Underline display

UN

Underline display

EVALID

The extended validation attribute for the field. This value is valid only when the EATTR value on the Update/Show Screen Parameters screen is Y and you request that CA Telon generate one of the following types of source code:

- IMS MFS, on the Update TSO/IMS Screen
- CICS BMS, on the Update CICS Screen Environment screen

Values are:

MF

The application user must enter all character positions of the field

ME

The application user must enter at least one character into the field

BO

Both of the above apply

During execution, terminal input in the field is checked before control is returned to the program. Control is returned to the program only if something is entered. By comparison, the REQ field value is used to check for input from inside the program. See Update Panel Fields (Online) for more information on the REQ field.

If the application uses terminals both with and without extended attributes and the EVALID is ME, the REQ value must be Y.

FMTEXIT

(IMS applications only.) An MFS field exit routine invoked for the field on input. Two values are required for this specification:

- Exit Routine Number (values: 000 through 127)
- Exit Routine Value (values: 000 through 255)

FMTCNTL

(IMS applications only.) A value to specify that this is an output field displaying special MFS data. Values are:

Y

This is a special MFS field

N

This is not a special MFS field

If FMTCNTL is Y, the FIELD NAME value (LABEL field value on other screens) must be one of the MFS system literals:

- DATE1
- DATE2
- DATE3
- DATE4
- TIME
- PASSWORD
- LTSEQ
- LTNAME
- LPAGENO
- LTMSG

For fields with labels other than PASSWORD, the field usage is assumed to be OUTPUT (other usage specifications are ignored). The field occupies space on the screen, but does not appear in the buffers (that is, the CA Telon program knows nothing about the field).

Field Definitions

COMMAND

For information, see Primary Commands.

FIELD NAME OR POSITION

(*Protected field.*) Displays the field name. See Panel Definition Menu or Update Panel Fields (Online) for more information.

IEXTEND

The extended parameter list for input edit. The list can contain up to nine parameters and can include any COBOL or PL/I data name defined to the program.

The parameters that you specify override the extra parameters (for example, extensions) passed to the user-written output field edit (defined earlier using the FLDTYPE field).

Note: CA Telon supports up to 300 override extensions per program.

OEXTEND

The extended parameter list for output edit. The list can contain up to nine parameters and can include any COBOL or PL/I data name defined to the program.

The parameters that you specify override the extra parameters (for example, extensions) passed to the user-written output field edit (defined earlier using the FLDTYPE field).

Note: CA Telon supports up to 300 override extensions per program.

Field Definitions

COMMAND

For information, see Primary Commands. You can also use the LOCATE command to find an item by its sequence number, using this syntax:
Locate sequence-number

SEQ

A field in which you can enter one of the following line commands:

C

Copy a line.

CC

Copy a block of lines

I

Insert a line

Inn

Insert *nn* lines

IS(*n*)

Insert *n* lines for you to enter the command

M

Move a line.

MM

Move a block of lines

R

Repeat a line.

RR

Repeat a block of lines.

A

Line(s) to insert, copy, or move go after this line.

B

Line(s) to insert, copy, or move go before this line.

D

Delete an entry.

U

Update an entry.

You are transferred to the appropriate screen:

- Update Cross-Field Edit (XFEDIT)
- Update Segment Edit (SEGEDIT)

See Line Commands for more information.

TYPE

The type of consistency edit that CA Telon is to perform. Values are:

- XFEDIT
- SEGEDIT
- SRC

See the *Programming Concepts Guide* for general information on SRC, XFEDIT, and SEGEDIT.

DESCRIPTION

The value of this field is:

- An identifier for and description of the consistency edit, if the TYPE value is XFEDIT or SEGEDIT
- The COBOL or PL/I source statement embedded in the code, if the TYPE value is SRC

This value cannot contain a single quote (') or an ampersand (&). If it does, errors will occur during program assembly.

For XFEDIT or SEGEDIT types, CA Telon uses the value to identify the edit and initialize the error message for the edit. The first blank-delimited word or the first eight bytes (whichever is shorter) becomes an identifier to permit direct reference to the edit. If the first word is less than 8 bytes, CA Telon right-justifies the remainder of the value to isolate the first word at the left of the display. If you enter **EDIT1 - THIS IS A SEGEDIT**, the display returned is:

```
1__ SEGEDIT EDIT1___ - THIS IS A SEGEDIT
```

The entire description is placed in a comment at the beginning of the section or procedure in the generated source.

If you enter an XFEDIT or SEGEDIT line on this screen, you must update the XFEDIT or SEGEDIT before editing on this screen.

See Update Cross-Field Edit (XFEDIT) and Update Cross-Field Edit (SEGEDIT) for more information.

Update Cross-Field Edit (XFEDIT)

Access

On the List Consistency Edits screen, enter **U** in the SEQ field and **XFEDIT** in the TYPE field.

After entering the data, save the entries by pressing the END PF key. If you entered **U** in the multiple occurrences of the SEQ field on the List Consistency Edits screen, the update screen for the next edit is displayed. Otherwise, control returns to the List Consistency Edits SEGEDIT screen.

Program ID

P165

Function

Updates cross-field edit data. See Consistency Edits Example for examples of entering cross-field edit fields.

```
HHNNNN.PD UPDATE XFEDIT *****  
COMMAND ==> _____  
  
                        EDIT NAME XXXXXXXX  
  
COPY EDIT BASE:  _____      SEGL00P:  _  
EDIT CONDITION:  _____  
                  _____  
                  _____  
                  _____  
                  _____  
ERROR MESSAGE:  _____  
                  _____  
HIGHLIGHT FIELDS: _____  
ERRCHAR FIELDS:  7 _____  
CURSOR AT FIELD:  8 _____
```

Field Definitions

COMMAND

For information, see Primary Commands.

EDIT NAME

(*Protected field.*) Displays the first word or first eight characters in the DESCRIPTION field on the List Consistency Edits screen.

COPY EDIT BASE

The name of an existing edit whose definition values you can copy into the fields on this screen with these steps:

1. Enter the EDIT NAME value of an edit that is listed on the List Consistency Edits screen
2. Press Enter.

All blank fields on the screen are initialized to the values associated with the edit you have copied.

EDIT CONDITION

A conditional test for fields. When the condition is true, there is an error.

A condition can be either a COBOL 88-level field or a field/application user combination where:

- The field is a COBOL or PL/I variable or literal. Non-numeric literals must be enclosed in double quotes. The maximum length is 256 bytes including double quotes.
- The operation is one of the following:
 - The reserved word NUMERIC
 - The reserved word NOT NUMERIC
 - Any COBOL or PL/I operand
 - An expression in the format *mnemonic value* or *mnemonic,value* where *mnemonic* is LT, LE, EQ, GE, GT, or NE, and *value* is a COBOL or PL/I variable, a literal (enclosed in double quotes), a COBOL reserved word, or an expression (enclosed in single quotes) with arithmetic symbols
- Valid connectors are AND, OR, and THENIF

Note: THENIF causes the condition that follows to be nested inside the first condition. You cannot specify an ELSE condition.

A valid value is at least one condition. Entering **IF** is optional; it is generated in the code whether it is specified or not.

For example, if you specify

```
EMPL-DOB,NE,SPACES,THENIF,EMPL-DOB,GT,  
      '(XFER-CURRENT-DATE - 160000)'
```

CA Telon generates this COBOL code:

```
IF EMPL-DOB NOT=SPACES  
  IF EMPL-DOB > (XFER-CURRENT-DATE - 160000)
```

ERROR MESSAGE

The error message that appears in the ERRMSG1 field when an edit condition is true. This value is initialized to the contents of the DESCRIPTION field on the List Consistency Edits screen.

You can specify a variable or a literal by enclosing the value in double quotes ("). If the value contains neither double quotes nor embedded blanks, it identifies a host variable name (HVNAME).

For example:

- If the value in this field is XFER-ERROR-MESSAGE, CA Telon generates this COBOL code:

```
MOVE XFER-ERROR-MESSAGE TO ERRMSG-FIELD.
```

- If the value in this field is "ERROR MESSAGE", CA Telon generates this COBOL code:

```
MOVE 'ERROR MESSAGE' TO ERRMSG-FIELD.
```

HIGHLIGHT FIELDS

The fields highlighted when the error condition is true. Values are input, output, or select field names defined in the NAME field on the Update Panel Fields (Online) screen. See Update Panel Fields (Online) for more information.

Note: CA Telon requires a value in at least one of these fields:

- HIGHLIGHT FIELDS
- ERRCHAR FIELDS
- CURSOR AT FIELD

ERRCHAR FIELDS

Required screen fields. Valid values are the names of input, outin, or select field for which the REQ value is C on the Update Output/Input/Outin Field screen.

When an application user fails to input a required field, the field is highlighted, flagged with the error-required character (usually '*'), and returned to the application user. Cross-field edits are done before the field is returned to the user.

If a field is specified here but its REQ value is not C, CA Telon automatically fills a field with zeros or spaces (depending on type) when an application user fails to input a value.

Note: CA Telon requires a value in at least one of these fields:

- HIGHLIGHT FIELDS
- ERRCHAR FIELDS
- CURSOR AT FIELD

CURSOR AT FIELD

The field where the cursor appears when the screen is returned to the application user with an error condition. Valid values are the names of input, outin, or select field names defined in the NAME field on the Update Output/Input/Outin Field screen.

If you do not enter a value, the cursor appears at the first highlighted field.

Note: CA Telon requires a value in at least one of these fields:

- HIGHLIGHT FIELDS
- ERRCHAR FIELDS
- CURSOR AT FIELD

SEGLOOP

Identifies the XFEDIT as belonging to a SEGLOOP. When a Y is entered in this field, CA Telon generates code to perform the cross-field edit in a DO-group, testing all iterations of the field in one pass, and putting out one error message for all fields in error.

Update Segment Edit (SEGEDIT)

Access

On the List Consistency Edits screen, enter **U** in the SEQ field next to the segment edit to be created or updated.

After entering the data, save the entries by pressing the END PF key. If you have entered **U** in multiple occurrences of the SEQ field of the List Consistency Edits screen, the update screen for the next edit is displayed. Otherwise, control returns to the List Consistency Edits screen.

Program ID

P168

Function

Updates segment edit data.

HHNNNN.PD UPDATE SEGEDIT *****	
COMMAND ==> _____	
EDIT NAME XXXXXXXX	
COPY EDIT BASE: _____	SEGLoop: _
SEGMENT NAME: _____	PCBNAME: _____
KEY: _____	
WHEN: _____	
ERROR CONDITION: _____	
HIGHLIGHT FIELDS: _____	
ERRCHAR FIELDS: _____	
CURSOR FIELD: _____	
ERROR MESSAGE: _____	
CALL FUNC: _____	OPCODE: _____
DLI QUALIFY _____	CMDCODE: _____ I/O AREA: _____
SSALIST: _____	
VSAM SEGMENT LTH: _____	
GEN KEY LTH: _____	

Field Definitions**COMMAND**

For more information, see Primary Commands in the chapter "Editors and Commands."

EDIT NAME

(*Protected field.*) Displays the first word or first eight characters in the DESCRIPTION field on the List Consistency Edits screen.

COPY EDIT BASE

The name of an existing edit whose definition values you can copy into the fields on this screen with these steps:

1. Enter the EDIT NAME value of an edit that is listed on the List Consistency Edits screen
2. Press Enter.

All blank fields on the screen are initialized to the values associated with the edit you have copied.

SEGMENT NAME

The name of the DL/I segment, VSAM data set, or SQL TLNROW being accessed by the SEGEDIT:

- For DL/I:
 - With segments in multiple PSBs, the value specified in the LABEL field on the Create/Update Data Group screen
 - Otherwise, the value specified in the SEGMENT field on the Update Database Segment screen
- For VSAM, the value specified with the DATA ITEM (data set) statement in the NAME or LABEL field on the Show/Update Data Set Default Data screen
- For SQL, the value specified in the CURRENT ROW NAME field on the Select New Row Name screen

KEY

The name of the COBOL or PL/I variable that holds the key identifying the record, segment, or row to be read. This value overrides the value in the KEY field on the Create/Update Data Group screen.

Alternatively, you can enter the entire SEGEDIT WHERE clause in this field. Use double quotes where single quotes should appear in the generated output.

For SQL, there must be a one-to-one correspondence between host variables listed here and the key fields defined for the SEGEDIT TLNROW in the data administration option of TDF. To use different key fields, set up a new TLNROW with only the keys referenced in the SEGEDIT key clause.

WHEN

A conditional test for execution of the segment edit, in this format:

condition1[,connector1,condition2...]

When the condition is true, the remainder of the segment edit is executed. If no value is specified, the segment edit is always executed.

Condition can be either a COBOL 88-level field or a field/application-user combination where:

- The field is a COBOL or PL/I variable or literal. Non-numeric literals must be enclosed in double quotes. The maximum length is 256 bytes including double quotes.
- The operation is one of the following:
 - The reserved word NUMERIC
 - The reserved word NOT NUMERIC
 - Any COBOL or PL/I operand
 - An expression in the format *mnemonic value* or *mnemonic,value* where *mnemonic* is LT, LE, EQ, GE, GT, or NE, and *value* is a COBOL or PL/I variable, a literal (enclosed in double quotes), a COBOL reserved word, or an expression (enclosed in single quotes) with arithmetic symbols

Connector can be AND, OR, and THENIF.

Note: THENIF causes the condition that follows to be nested inside the first condition. You cannot specify an ELSE condition.

Entering **IF** is optional; it is generated in the code whether it is specified or not.

ERROR CONDITION FOUND/ NOTFOUND

The condition that causes the execution of the error routine and redisplay of the screen.

A FOUND condition exists when the:

- DL/I return status code is spaces
- VSAM return condition is nulls
- SQL return code is 000
- First byte of TS or TD queue is null (a read to a TD queue record deletes the record)

A NOT FOUND condition exists when the:

- DL/I return status code is GE
- VSAM return code is NOTFND
- SQL return code is +100
- TS queue return status code is ITEMERR
- TD queue return status code is QZERO

Note: Some SQL databases may use non-standard return codes for FOUND and NOT FOUND conditions. The return codes for these databases are converted to standard SQL return codes prior to use within generated programs.

HIGHLIGHT FIELDS

The fields to be highlighted when the error condition is true. Valid values are input, output, or select field names defined in the NAME field on the Update Panel Fields (Online) screen.

See Update Panel Fields (Online) for more information.

Note: CA Telon requires a value in at least one of these fields:

- HIGHLIGHT FIELDS
- ERRCHAR FIELDS
- CURSOR AT FIELD

ERRCHAR FIELDS

Required screen fields. Valid values are the names of input, outin, or select field for which the REQ value is C on the Update Output/Input/Outin Field screen.

When an application user fails to input a required field, the field is highlighted, flagged with the error-required character (usually '*'), and returned to the application user. Cross-field edits are done before the field is returned to the user.

Note: If a field is specified here but its REQ value is not C, CA Telon automatically fills a field with zeros or spaces (depending on type) when an application user fails to input a value.

CA Telon requires a value in at least one of these fields:

- HIGHLIGHT FIELDS
- ERRCHAR FIELDS
- CURSOR AT FIELD

CURSOR AT FIELD

The field where the cursor appears when the screen is returned to the application user with an error condition. Valid values are the names of input, outin, or select field names defined in the NAME field on the Update Output/Input/Outin Field screen.

If you do not enter a value, the cursor appears at the first highlighted field.

Note: CA Telon requires a value in at least one of these fields:

- HIGHLIGHT FIELDS
- ERRCHAR FIELDS
- CURSOR AT FIELD

ERROR MESSAGE

The error message that appears in the ERRMSG1 field when an edit condition is true. This value is initialized to the contents of the DESCRIPTION field on the List Consistency Edits screen.

You can specify a variable or a literal by enclosing the value in double quotes ("). If the value contains neither double quotes nor embedded blanks, it identifies a host variable name (HVNAME).

For example:

- If the value in this field is XFER-ERROR-MESSAGE, CA Telon generates this COBOL code:

`MOVE XFER-ERROR-MESSAGE TO ERRMSG-FIELD.`
- If the value in this field is "ERROR MESSAGE", CA Telon generates this COBOL code:

`MOVE 'ERROR MESSAGE' TO ERRMSG-FIELD.`

CALL FUNC

The I/O access code. Values are:

DLI

Any four-byte GET function code, such as GN or GHU

VSAM

Any eight-byte READ verb, such as READ or READNEXT

SQL

(This option not valid for SQL)

OPCODE

The operation code. This value overrides the value specified in the OP field on the Update Database Segment screen. If not specified, the default is **>=** for BROWSE segments and **=** for all other segment types.

For DL/I processing, this value causes CA Telon to change the SSA before the call and reset it after the call. The OPCODE field and the SSALIST field are mutually exclusive.

For VSAM processing, the only values are **>=** and **=**.

For SQL processing, valid relational operators are allowed. The default is **=**.

DLI QUALIFY

The DL/I SSA for the segment edit. Values are:

Y

Qualify the SSA (*segment-QUAL-SSA*)

N

Do not qualify the SSA (*segment-UNQUAL-SSA*)

Note: The value in this field and the value in the SSALIST field are mutually exclusive; specify only one.

CMDCODE

The DL/I SSA command code. This value overrides the command code value, if specified, in the CMND field on the Update Database Segment screen.

The character(s) you specify are prefixed with an asterisk (*) and suffixed with one or two dashes (-) to make a four-byte code.

Note: The value in this field and the value in the SSALIST field are mutually exclusive; specify only one.

I/O AREA

The name of the data area used for this edit. This value overrides the value specified in the I/O AREA field on the Update DL/I Detail Data Access screen or the CA Telon generated I/O area.

SSALIST

The list of DL/I SSAs. It overrides SSAs generated by CA Telon from DSCREF field values on the Update Database Segment screen.

You can include the current SSAs in this list. CA Telon assumes these SSAs are defined in your custom code.

VSAM SEGMENT LTH

The maximum length of VSAM variable-length records. Valid values include an integer or the name of a variable that contains the length.

This value overrides the value specified in the RECLTH field on the Show/Update Data Set Default Data screen.

GEN KEY LTH

(VSAM processing only.) The length of the generic key used for the access. Valid values include either an integer or the name of a COBOL or PL/I variable that contains the key length value.

If you do not enter a value, CA Telon assumes that the access uses the full key length.

PCBNAME

The PCB against which the segment edit is to execute.

SEGLOOP

Identifies the SEGEDIT as belonging to a SEGLOOP. When a Y is entered in this field, CA Telon generates code to perform the cross-field edit in a DO-group, testing all iterations of the field in one pass, and putting out one error message for all fields in error.

Consistency Edits Example

Access

This scenario shows the entries an application developer would make to a panel definition to perform these consistency edits:

- Make sure the student enrollment data is not greater than today's date
- Perform a section/procedure named MY-CUSTOM-CODE-SECTION
- Check student number against the DL/I database to make sure the number does not already exist

Values entered by the application developer are identified in the text and highlighted on the screen illustration.

Step 1

The application developer accesses the Panel Definition menu and enters these values:

- **CR** in the FUNCTION field
- **CE** in the ITEM field

- **ED** in the HEADER field
- **MENU ADD STUDENT** in the ID field

Note: ADDSTU is displayed in the FIELD field because this edit is for a select field. Otherwise, *PANEL would be displayed in this field.

```

PANEL DEFINITION MENU *****
COMMAND ==> _____

FUNCTION: CR    CR-CREATE    UP-UPDATE    PU-PURGE    SH-SHOW    LI-LIST

ITEM      CE    PI-IMAGE    PD-DEFIN
                FD-FIELD    CE-CONSIG    SL-SEGL00P
                (UP)        (CR,UP)      (CR,UP,PU)

MEMBER NAME:
  HEADER  ED
  ID      MENU ADD STUDENT
  DESC    _____

ENTER VALUE FOR SPECIFIC ITEM TO BE PROCESSED:
  1. IMAGE      < > + | \ (INPUT OUTPUT OUTIN SELECT LIT-BREAK CHARACTERS)
                24 080 (LINE-COLUMN IMAGE SIZE)
                U      (UPPER/LOWER CASE LITERALS)

  2. DEFIN      Y Y Y Y N (INPUT OUTPUT OUTIN SELECT LITERAL FIELDS LISTED)
  3. FIELD      ADDSTU___ (NAME OR LINE,COLUMN OR "*PANEL")
  4. CONSIG     _____ (TYPE - "XFEDIT", "SEGEDIT", OR BLANK FOR LIST)
                _____ (NAME - IF TYPE SPECIFIED)
                _____ (TYPE - "FILE" OR "TABLE")
                _____ (FROM NAME OR LINE,COLUMN)
                _____ (TO NAME OR LINE,COLUMN)

```

Step 2

After the application developer presses Enter, the List Consistency Edits screen is displayed.

In this example, two consistency edits (lines 1 and 3) and one line of source code (line 2) are defined for the field. The application developer enters **U** as line commands for lines 1 and 3 to request update of the two consistency edits.

```
EDMENU.PD LIST SRC, XFEDIT, SEGEDIT* *****  
COMMAND ==> _____ PAGE 01  
  
SEQ  TYPE  DESCRIPTION (FIRST WORD IS XFEDIT/SEGEDIT NAME) OR STATEMENT CODE  
U01  XFEDIT DATECHK - MAKE SURE ENROLLMENT DATE IS VALID  
002  SRC    PERFORM MY-CUSTOM-CODE-SECTION.  
U03  SEGEDIT STUNOCK - MAKE SURE STUDENT NO IS NEW  
004  _____  
005  _____  
006  _____  
007  _____  
008  _____  
009  _____  
010  _____  
011  _____  
012  _____  
013  _____  
014  _____  
015  _____  
016  _____  
017  _____  
018  _____
```


Step 3

After the application developer presses Enter, the Update Cross-Field Edit (XFEDIT) screen is displayed.

The application developer modifies the EDIT CONDITION field and adds a value to the HIGHLIGHT FIELDS field to identify the field for highlighting if the application user enters data that is not consistent with the edit.

```
EDMENU.AD UPDATE XFEDIT *****
COMMAND ==> _____

                        EDIT NAME DATECHK

COPY EDIT BASE: _____          SELOOP:  _
EDIT CONDITION:  STUDENT-ENROLL-DATE GT XFER-TODAYS-DATE_____
                _____
                _____
                _____
ERROR MESSAGE:   DATECHK MAKE SURE ENROLLMENT DATE IS VALID
HIGHLIGHT FIELDS: ENROLLDT_____
ERRCHAR FIELDS:  _____
CURSOR AT FIELD: _____
```

Step 4

The application developer presses End to save this screen and display the Update Segment Edit (SEGEDIT) screen for the second edit, STUNOCK, previously selected on the List Consistency Edits screen.

EDMENU.PD UPDATE SEGEDIT *****			
COMMAND ==> _____			
EDIT NAME STUNOCK			
COPY EDIT BASE:	_____	SEGL00P:	__
SEGMENT NAME:	_____	PSBNAME	_____
SEGKEY:	_____		
WHEN:	_____		

ERROR CONDITION:	_____		
HIGHLIGHT FIELDS:	_____		
ERRCHAR FIELDS:	_____		
CURSOR FIELD:	_____		
ERROR MESSAGE:	STUNOCK MAKE SURE STUDENT NO IS NEW _____		

CALL FUNC:	_____	OPCODE:	_____
DLI QUALIFY:	_____	CMDCODE:	_____
SSALIST:	_____	I/O AREA:	_____

VSAM SEGMENT LTH:	_____		
GEN KEY LTH:	_____		

The application developer adds or modifies these values:

- **EDSTUDNT** in the SEGMENT NAME field, to identify the DL/I segment for accessing
- **TPI-IDENT** in the SEGKEY field. The input value is IDENT and CA Telon adds the prefix TPI-
- **'TPI-OPTION= "1"'** to provide the condition, to be generated in an IF statement, for execution of this edit identify the input buffer field containing the segment key
- **FOUND** in the ERROR CONDITION field, to identify the condition of the error process (that is, the error condition exists whenever the record is found)

- **IDENT** in the HIGHLIGHT FIELDS field, to identify the field to highlight whenever the error is processed
- **STUDENT ID IS ALREADY ON DB** replaces the default error message that had been carried over from the List Consistency Edits screen

```

EDMENU.PD UPDATE SEGEDIT *****
COMMAND ==> _____
                                EDIT NAME STUNOCK
COPY EDIT BASE: _____
SEGMENT NAME:  EDSTUDNT        PSBNAME _____
SEGKEY:        TPI-IDENT
WHEN:          'TPI-OPTION="1"'
               _____
               _____
               _____
ERROR CONDITION: FOUND
HIGHLIGHT FIELDS: IDENT
ERRCHAR FIELDS: _____
CURSOR FIELD:   _____
ERROR MESSAGE:  STUDENT ID IS ALREADY ON DB
               _____
               _____
CALL FUNC:      _____  OPCODE: _____
DLI  QUALIFY:   _____  CMDCODE: _____  I/O AREA: _____
    SSALIST:    _____
VSAM SEGMENT LTH: _____
    GEN KEY LTH: _____
  
```

Step 5

When the application developer presses End, this screen is saved and the List Consistency Edits screen is redisplayed.

Create/Update Table SEGLOOP

Access

On the Panel Definition menu, enter:

- **CR** or **UP** in the FUNCTION field
- **SL** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field
- **TABLE** in the SEGLOOP TYPE field
- *Field-name* or field position in the SEGLOOP FROM field, if FUNCTION is CR
- *Field-name* or field position in the SEGLOOP TO field, if FUNCTION is CR

Program ID

P170

Function

Create or update data for a table segment loop. A table segment loop retrieves the information to be displayed from a table rather than by browsing a segment, as is done with a file segment loop.

```
HHIIII.PD CREATE TABLE SEGLOOP *****
COMMAND ==> _____

                _____ SEGLOOP   LIMITS   NAME   LINE   COLUMN
                        FIRST
                        LAST   _____   ___   ___

INCRE _____
REPEAT _____
CINCRE _____
LINECNT _____ COLSGLP _____
OCUST1 _____ OCUST2 _____ OCUST3 _____
OSEGIDX _____
SAVEKEY _____ TO _____
                        TO _____
                        TO _____
                        TO _____
                        TO _____
                        TO _____
                        TO _____
                        TO _____

ISEGIDX _____
ICUST1 _____ ICUST2 _____ ICTLNM _____
```

Field Definitions

COMMAND

For information, see Primary Commands.

SEGLOOP

The type of SEGLOOP. Possible values are OUTPUT, INPUT, and OUTIN. The value is determined automatically by the type of fields in the loop range.

FIRST

First field name in the SEGLOOP. This value is initialized to the value in the FROM field on the Panel Definition menu. Names are limited to six characters for IMS and five characters for CICS.

LAST

Last field name in the SEGLOOP. This value is initialized to the value in the TO field on the Panel Definition menu. Names are limited to six characters for IMS and five characters for CICS.

LINE

The line number positions of the first field and the last fields in the SEGLOOP group.

COLUMN

The column number positions of the first field and the last fields in the SEGLOOP group.

INCRE

The vertical spacing for displaying the group of SEGLOOP fields on the screen, specified in this format:

vertical-spacing1[,vertical-spacing2...]

The vertical starting point of the first line displayed is specified in the combination of the FIRST and LINE fields, and the value in the INCRE field is the offset relative to the position of the first line displayed. The number of offset values specified must be one less than the number of lines that comprise the display.

For example, if the first field is FLD1, its LINE value is 3, its COLUMN value is 6, and the INCRE specification is 1,2,1, the vertical spacing of the resulting display is:

```
1---+---+---+---+
001
002
003  FLD1
004  FLD1
005
006  FLD1
007  FLD1
```

Note: If you want to display a specific number of iterations, you specify one less entry. In the example above, you specify three entries to display four iterations.

REPEAT

The number of times to repeat the display defined by the INCRE value.

CINCRE

Define the spacing between columns on the screen; that is, the number of columns between one field in an iteration and the next iteration of the same field. (A value of 15 would cause a field in the next loop to begin 15 columns to the right of the same field in the prior loop, regardless of the other fields in the loop.) Specify one CINCRE value for each column, except for the first.

For example, if you want data to appear in three columns, with the first column beginning at position 2, the second beginning in position 20, and third beginning at position 45, enter the CINCRE values:

18, 25

(that is, from position 2 to position 20 is 18 columns; from position 20 to 45 is 25 columns).

The number of values specified in the CINCRE field determines the number of iterations of the line. That is, if you specify n values in the number of iterations on each line is $n+1$.

For BMS processing, when you specify CINCRE, all fields in SEGLOOP must appear on one line only. For non-BMS processing, when you specify CINCRE, all output and input fields in SEGLOOP must occur on one line only.

Note: When coding CINCRE, be sure that the values you specify allow enough spaces to display all fields in the loop horizontally. Also, be sure that the number of columns needed to display each line does not exceed the width of the screen.

LINECNT

The name of an output field in which to display the line number. To suppress display of the line number, do not enter a value here.

COLSGLP

Specify whether you want to display a multi-column SEGLOOP in column instead of row order. Only valid if CINCRE is specified. Values are:

N

(Default.) Display multi-column SEGLOOP in row order

Y

Display multi-column SEGLOOP in column order

OCUST1

Custom code name used to perform output processing after the initial auto exec call and before mapping the first line to the screen.

OCUST2

Custom code name used to perform output processing after the second auto exec call in the SEGMENT loop and before the data is mapped to the screen. CA Telon performs this code if the call is successful.

Note: For paging SEGLOOPS, CA Telon performs one additional read after all SEGLOOP iterations on the screen have been filled to establish a starting point for the next page. Use OCUST2 to check additional validation criteria for display. This precludes display of a blank page when a successful auto exec data access request is not sufficient for validation.

OCUST3

Custom code name used to perform output processing at the end of the SEGMENT loop. CA Telon uses it for data that is mapped to the next line of the current screen during the next loop iteration.

CA Telon does not execute this code if there is no more room on the screen to contain the information mapped from the file or if the auto exec call is unsuccessful.

OSEGIDX

Name of the index for the array from which output values are mapped for each SEGLOOP pass. CA Telon uses this information only if values are mapped from an array. No specification is needed if the subscript SEGLOOP-COUNT is to be used instead of an index name.

SAVEKEY

Names of the data items that are stored in a table at each loop iteration. The specification is one or more pairs of data items, each including a source field and a destination field. The second data item specified is the name of a table element (generally in the transfer work area) in which to store information from the first data item in each iteration of the SEGLOOP process.

The maximum length of this field is 254 characters including delimiters.

Generally, the data items define the keys for the displayed data, as specified in the DBNAME field on the Update Output/Input/Outin Field screen, and are used to retrieve a requested item (defined by line number) during subsequent select-option processing.

ISEGIDX

Name of the index for the array being referenced in each segment loop. CA Telon uses this information only if values are mapped into an array. No specification is needed if the subscript SEGLOOP-COUNT is used instead of an index name.

ICUST1

Name of the exit for custom code prior to input edits (in the loop).

ICUST2

Name of the custom code used to perform custom processing in the input SEGMENT loop.

ICTLNM

Screen field name, in the input segment loop, that determines if editing or mapping is done on the current iteration of the loop.

If the specified screen field is blank, CA Telon:

- Sets the LINE-INPUT-ERROR field value to N
- Does not increment the LINE-INPUT-COUNT value
- Passes control to custom code (if ICUST2 is specified) or to the bottom of the input loop; optionally, you can terminate the processing of the loop in the custom code

Create/Update File SEGLOOP

Access

On the Panel Definition menu, enter:

- **CR** or **UP** in the FUNCTION field
- **SL** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field
- **FILE** in the SEGLOOP TYPE field
- *Field-name* or field position in the SEGLOOP FROM field, if FUNCTION is CR
- *Field-name* or field position in the SEGLOOP TO field, if FUNCTION is CR

Program ID

P175

Function

Create or update data for a file segment loop. A file segment loop retrieves information by browsing a segment, file, or criteria from table SELECT, rather than from a table, as is done with a table segment loop.

HHIIII.PD UPDATE SEGLOOP *****									
COMMAND ==> _____									
				SEGLOOP		LIMITS	NAME	LINE	COLUMN
						FIRST			
						LAST			
INCRE									
REPEAT									
CINCRE									
LINECNT	SEQ			COLSGLP					
PAGE	PAGESAV			PKYUNIQ		PKYLTH			
PAGEKEY	EMPL-ID								
OCUST1		OCUST2			OCUST3				
OSEGIDX									
SAVEKEY									
						TO			
						TO			
						TO			
						TO			
						TO			
						TO			
						TO			
STBRKEY						ISEGIDX			
ICUST1		ICUST2			ICTLNM				

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

SEGLOOP

Type of SEGLOOP. Possible values are OUTPUT, INPUT, and OUTIN. The value is determined automatically by the type of fields in the loop range.

FIRST

First field name in the SEGLOOP. This value is initialized to the value in the FROM field on the Panel Definition menu. Names are limited to six characters for IMS and five characters for CICS.

LAST

Last field name in the SEGLOOP. This value is initialized to the value in the TO field on the Panel Definition menu. Names are limited to six characters for IMS and five characters for CICS.

LINE

Line number of the field.

COLUMN

Field column number.

INCRE

Vertical spacing for displaying the group of SEGLOOP fields on the screen; specified in the following format:

vertical-spacing1[,vertical-spacing2...]

The vertical starting point of the first line displayed is specified in the combination of the FIRST and LINE fields, and the value in the INCRE field is the offset relative to the position of the first line displayed. The number of offset values specified must be one less than the number of lines that comprise the display.

For example, if the first field is FLD1, its LINE value is 3, its COLUMN value is 6, and the INCRE specification is 1,2,1, the vertical spacing of the resulting display is:

1---+---+---+---+

001

002

003 FLD1

004 FLD1

005

006 FLD1

007 FLD1

Note: If you want to display a specific number of iterations, you specify one less entry. In the example above, you specify three entries to display four iterations.

REPEAT

Specify the number of times you want to REPEAT the INCRE in the group. For example, a repetition of four results in four occurrences of the INCRE, which results in four plus the one occurrence of the group that was painted on the Panel Image.

CINCRE

horizontal-spacing1 [, horizontal-spacing2]...

Define the spacing between columns on the screen; that is, the number of columns between one field in an iteration and the next iteration of the same field. (A value of 15 would cause a field in the next loop to begin 15 columns to the right of the same field in the prior loop, regardless of the other fields in the loop.) Specify one CINCRE value for each column, except for the first.

For example, if you want data to appear in three columns, with the first column beginning at position 2, the second beginning in position 20, and third beginning at position 45, enter the CINCRE values:

18, 25

(that is, from position 2 to position 20 is 18 columns; from position 20 to 45 is 25 columns).

The number of values specified in the CINCRE field determines the number of iterations of the line. That is, if you specify n values in CINCRE, the number of iterations on each line is $n+1$.

For BMS processing, when you specify CINCRE, all fields in SEGLOOP must appear on one line only. For non-BMS processing, when you specify CINCRE, all output and input fields in SEGLOOP must occur on one line only.

Note: When coding **CINCRE**, be sure that the values you specify allow enough spaces to display all fields in the loop horizontally. Also, be sure that the number of columns needed to display each line does not exceed the width of the screen.

LINECNT

The name of an output field which is to display the line number. To suppress display of the line number, do not enter a value.

COLSGLP

Specify whether you want to display a multi-column SEGLOOP in column instead of row order. Only valid if CINCRE is specified. Values are:

Y

Display multi-column SEGLOOP in column order

N

Display multi-column SEGLOOP in row order

PAGE

A value to specify whether CA Telon or the application user is to generate paging. Values are:

Y

Automatically generate paging

N

Do not automatically generate paging

PAGESAV

The number of screens that can be paged backward consecutively before the first screen is redisplayed.

PKYUNIQ

A value to specify whether non-unique keys are allowed for the PAGEKEY field. Values are:

Y

Key must be unique

N

Non-unique keys allowed

PKYLTH

The length (in bytes) of the key identified by the PAGEKEY value.

PAGEKEY

The data item that contains the key of the segments being displayed on the list screen. This data is saved in the transfer work area to implement automatic paging.

Note: For SQL, a PAGEKEY value is not valid. If you specify a value in this field, the Generator issues an error message.

OCUST1

The name of the custom code to perform output processing after the initial auto exec call and before the first line mapped to the screen. OCUST1 custom code is usually used in tandem with custom code identified in either the OCUST2 field or OCUST3 field.

OCUST2

The name of the custom code to perform output processing after all subsequent auto exec calls in the segment loop, before the data is mapped to the screen. CA Telon performs this code if the auto exec call is successful, whether or not the end of page has been reached.

Note: CA Telon performs one additional read after all SEGLOOP iterations have been filled, to establish a starting point for the next page. Use OCUST2 to check for additional validation criteria for display. This precludes display of a blank page when a successful auto exec call is not sufficient for validation.

OCUST3

The name of the custom code to perform output processing at the end of the segment loop. CA Telon uses this code for data that is mapped out to the next line of the current screen during the next loop iteration. CA Telon does not execute this code if there is no more room on the screen to contain the information mapped from the file.

OCUST3 and OCUST1 for the first record, are generally used to format data for display on the screen.

OSEGIDX

The name of the index for the array from which output values are mapped for each SEGLOOP pass. This information is used only if values are mapped from an array. No specification is needed if the subscript SEGLOOP-COUNT is used instead of an index name.

SAVEKEY

The names of the data items that are stored in a table at each loop iteration. The specification is one or more pairs of data items, each including a source field and a destination field. The second data item specified is the name of a table element (generally in the transfer work area) in which to store information from the first data item in each iteration of the SEGLOOP process.

The maximum length of this field is 254 characters including delimiters.

Generally, the data items define the keys for the displayed data, as specified in the DBNAME field on the Update Output/Input/Outin Field screen, and are used to retrieve a requested item (defined by line number) during subsequent select-option processing.

Note: For TS queues, the key should be the queue ITEM number.

STBRKEY

The start browse key is the data item that contains the key data used in accessing the first record to appear in the SEGLOOP on this screen. Only a single item may appear in this field, though it may identify a structure (for COBOL) or the overlay of a structure (for PL/I).

For DL/I, CA Telon uses this value in the SSA for the first GU call to start the looping process. You cannot use this field with the SCHFLDC, SCHFLDI, and SCHFLDL fields.

For VSAM, CA Telon uses this value as a unique key that is read to start the looping process. For temporary storage queues, CA Telon uses it as the startup ITEM member for the browse. The value must be a binary half word.

For SQL, CA Telon uses the value in STBRKEY to assign to the aggregate structure CURRENT-SEGMENT-KEY. The individual components of that structure are subsequently assigned to the key fields used in the SQL SELECT call.

For PL/I, if the SEGLOOP key is defined as decimal (for example, PIC S9(5) COMP-3, DEC FIXED, SQL integer) or binary (for example, PIC S9(4) COMP, BIN FIXED, SQL float), you must define a character string to overlay the key field, and then define that overlay field as the STBRKEY. For example:

```
SEGLOOP ...STBRKEY=OVERDEC...
```

COBOL

```
05 DECFLD    PIC S9(5) COMP-3.  
05 OVERDEC   PIC X(3)  REDEFINES DECFLD.
```

PL/I

```
DCL DECFLD    DEC FIXED (6,0)  
OVERDEC       CHAR(4) BASED(ADDR(DECFLD));
```

For COBOL and PL/I DL/I programs, where the AUTOEXEC BROWSE segment has a KEYPIC on it, STBRKEY has the same definition as the KEYPIC field. See the *Programming Concepts Guide* for more information.

ISEGIDX

The name of the index for the array being referenced in each segment loop. CA Telon uses this information only if values are mapped into an array. No specification is needed if the subscript SEGLOOP-COUNT is to be used instead of an index name.

ICUST1

Exit name for custom code prior to input edits (in the loop).

ICUST2

Custom code name used to perform custom processing in the input segment loop.

ICTLNM

Screen field name, in the input segment loop, that determines if editing or mapping is done on the current iteration of the loop.

If the specified screen field is blank, CA Telon:

- Sets the LINE-INPUT-ERROR field value to N
- Does not increment the LINE-INPUT-COUNT value
- Passes control to custom code (if ICUST2 is specified) or to the bottom of the input loop; optionally, you can terminate the processing of the loop in the custom code

SCHFLDC

(Applicable only to a SEGLOOP program converted from a release prior to 2.0. Valid only for DL/I-controlled SEGLOOPS.) The data item that contains the search criteria used in accessing records on the list screen. This data is used in the SSA to determine which segments are listed and must be set up prior to the transfer to this screen.

This field is generally used to list all segments that have an IMS-defined search field equal to a particular value.

If you specify a value, be sure to enter = in the OPCODE field on the Update Database Segment screen for the lowest level browse segment to override the standard >= value.

SCHFLDC is not used when the STBRKEY field is used to set up the SSA for the search I/O. If SCHFLDC is specified, SCHFLDI and SCHFLDL values must also be specified.

SCHFLDI

(Applicable only to a SEGLOOP program converted from a release prior to 2.0. Valid only for DL/I-controlled SEGLOOPS.) The name of the field in the segment against which the SEGLOOP search is directed. The segments listed are those segments whose SCHFLDI field values match the value in the SCHFLDC field.

If you specify SCHFLDI, you must also specify SCHFLDC and SCHFLDL values.

SCHFLDL

(Applicable only to a SEGLOOP program converted from a release prior to 2.0. Valid only for DL/I-controlled SEGLOOPs.) The length of the SCHFLDC and SCHFLDI fields. CA Telon uses this value to set up the SSA for the search I/O.

If SCHFLDL is specified, SCHFLDC and SCHFLDI values must also be specified.

[illegible]

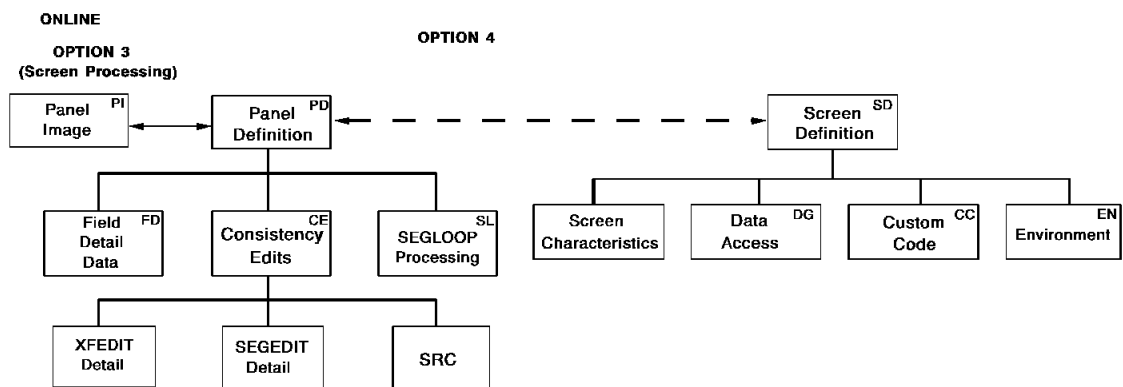
Chapter 6: Online Program Definition

Once you have created a panel image and panel definition, the next step in designing the program is to create a program definition using Option 4 on the TDF Main menu. That definition can be one of the following:

- Screen
- Driver
- Report
- Nonterminal program (CICS)

Note: The batch program definition function is described in Chapter 8, Prototyping Facility.

The following diagram shows the components of an online program definition:



Functionality

The online program definition function allows you to:

- Define general screen/report characteristics
- Define data accessed by the program
- Create custom code
- Identify the environment in which the program executes, including client/server generation.

Note: Using CA Telon definition utilities, previously-exported online program definitions can be imported into TDF online program definition for maintenance. See the *Utilities Guide* for more information.

Screen and Report Characteristics

The screen definition consists of field values the TDF requires to create your application program. For example, custom code created on other TDF screens is part of the screen definition.

The screen or report definition includes:

- Cursor positioning
- Next program to be executed
- Remarks
- Name of the transfer work area (XFERWKA) layout
- Name of the other program work area custom code
- Names of the various other custom code members added to designated areas of the program

Data Access

You can define two types of data access: automatically executed (auto exec) and user executed (user exec).

Auto exec I/O

When your program always performs the same action with the data (read, write, create, or update), you can create high-level, auto exec I/O. The CA Telon-generated program then accesses that data function with each execution of the program. Use the Create/Update Data Group screen to set up this automatic I/O.

CA Telon places code for auto exec calls in A-100-, B-100-, D-100-, and H-100- sections of the program depending on the type of call requested.

User exec I/O

When the same I/O must perform different processing based on varying conditions in the program, you create a user exec I/O with the Create/Update Data Group screen.

When the type of data access is conditional, you will create different I/O and control its execution. When you create user exec data access, CA Telon generates the code in a U-100 section or procedure that can be performed or called from any custom code entry point.

Custom Code

Although CA Telon automatically generates most of the code needed in your program, there are some cases that require your own custom code to complete your task.

Defining custom code

The custom code screen of the program definition submenu allows you to enter that code using the CA Telon editor. See the *Programming Concepts Guide* for program flowcharts that illustrate the normal entry points for your custom code.

Placing custom code

You can insert custom code at almost any point of your CA Telon-generated program. You identify the location for placing particular custom code members by naming those members in the screen definition, report definition, and driver definition.

Program Environment

Finally, you identify the environment in which the program runs. Screens described in this chapter allow you to define IMS/DC, CICS, or TSO (for test only) environment.

For the PWS environment, see the *PWS Option Administrator Guide*.

Online Program Definition Menu

Access

Access this menu in one of these ways:

- On the TDF Main menu, enter **4** in the FUNCTION field
- On most other CA Telon screens, enter **=4** in the COMMAND field

Program ID

S100

Function

Allows you to:

- Create, update, purge, show, and list program definitions
- Access the data group, custom code, and environment screens

This is the long form of the menu. CA Telon displays this menu when the USER MODE field on the Update Session Controls screen contains the value 2. CA Telon displays the short form of the menu when the USER MODE field on the Update Session Controls screen contains the value 1. See Update Session Controls for more information.

The short form of the menu contains a subset of the fields on the long form.

```

ONLINE PROGRAM DEFINITION MENU *****
COMMAND ==> _____

FUNCTION:  __  CR-CREATE  UP-UPDATE  PU-PURGE  SH-SHOW  LI-LIST
ITEM:      __  SD-SCREEN  DR-DRIVER  RD-REPORT  ND-NONTERMINAL
              DG-DATA GROUP  CC-CUSTOM CODE  EN-ENVIRON

MEMBER NAME:
HEADER  _____
ID      _____  TYPE __ (SD, DR, RD, ND)
DESC    _____

BASE DEFN :  _____  (FOR CREATE - NAME OF BASE SD, DR, RD OR ND)

ENTER VALUE FOR SPECIFIC ITEM TO BE PROCESSED:
1. ENVIRON  _____  (CICS, IMS)
2. CUSTCODE  _____  (NAME OF CUSTOM CODE)
  
```

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands." You can also enter the SETMODE command to swap between the long form and short form of this menu.

FUNCTION

The type of function to perform on the named ITEM. The initial value is UP (update). Values are:

CR

Create a screen definition, driver definition, report or CICS nonterminal definition.

UP

Update a screen definition, driver definition, report or CICS nonterminal definition.

PU

Purge a screen definition, driver definition, report or CICS nonterminal definition.

SH

Show a screen definition, driver definition, or report definition. The screen that is displayed consists of protected fields.

LI

List entities starting with the requested HEADER and ID. You can copy, rename, purge, update, or browse entities that are listed.

ITEM

The item with which you will work. Values are:

SD

Screen definition

DR

Driver definition

RD

Report definition

ND

CICS nonterminal definition

DG

Data group

CC

Custom code

EN

Environment

HEADER

A one- to five-character name identifying a group of programs (that is, an application or a portion of an application).

The length of this field is determined at installation time. The combined length of HEADER and ID must be five or six characters, depending on the installation options.

ID

A one- to five-character name that uniquely identifies one of the following within a particular HEADER:

- Screen definition
- Report definition
- CICS nonterminal definition
- Driver definition
- Panel definition
- Panel image

The combined length of HEADER and ID must be five or six characters, depending on the installation options.

TYPE

The type of definition. Values are:

SD

Screen definition

DR

IMS/DC Driver definition

RD

IMS/DC Report definition

ND

CICS nonterminal definition

DESC

The description entered on the Panel Definition menu. If you have not previously accessed the Panel Definition menu, this field is blank.

BASE DEFN

(Optional.) The concatenation of HEADER and ID values of an existing screen, driver, or report definition to copy.

Use this field only when the FUNCTION value is CR.

ENVIRON

The program environment targets. Values are:

- IMS
- CICS
- TSO (test environment only)

CUSTCODE

Name of the custom code member you are creating or updating. You must enter a value for this field when the value in the ITEM field is CC.

Create/Update Screen Definition

Access

On the Online Program Definition menu, enter:

- **CR** or **UP** in the FUNCTION field
- **SD** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field

You can also access this screen from the List Data Administration Information screen by entering **U** as a line command for a listing of a screen definition.

Program ID

S110

Function

Maintains characteristics of a screen definition and provides access to other TDF screens to complete the screen definition.

```

HHNNNN.SD UPDATE SCREEN DEFINITION ** *****
COMMAND ==>
OPTIONS ==> CUSTOM CODE _ DATA GROUP _ PANEL DEF _ ENV IMS _ SCRIN PARMS _
            STORED PROCEDURES _
GENERAL: DESC
*   NEXTPGM _ CURSOR _ SIZE _ X _ A REMARKS
*   CMPLOPT _ A IDENTIF _ A PROCEDR _
DATA   XFERWKA
AREAS: A WKAREA

OUTPUT:
A-100 A OINIT1 _ A OINIT2 _ A CURSCUS _
B-100 A OUTTERM _

INPUT:
P-100 PFKEYS
D-100 A ININIT1 _ A ININIT2 _
J-100 SELECT FIELDS
E-100 A FLDEDIT
X-100 _ SCREEN XFEDIT/SEGEDIT A CONSIS _
H-100 A INTERM _

MISC: A SECTION _
*     PGMCLST _

```

Fields allowing entry of multiple members

Five fields allow you to specify more than one member name:

- XFERWKA
- WKAREA
- PFKEYS
- SECTION
- PGMCUST

The XFERWKA, PFKEYS, and PGMCUST fields have no edit option field; therefore, you cannot select the custom code editor for these fields. Members entered into these fields are usually stored in shared libraries.

The WKAREA and SECTION fields have an edit option field. However, you can access the Editor only when one member name exists for the field. If you have specified more than one member name for the field, go to the List/Show Custom Code screen to select the member for editing.

All five fields that accept specification of multiple members can contain a maximum of 253 bytes of data. You can specify 60 bytes for each field on this screen; you can enter **U** in a one-byte field to the right of the 60th byte to request an extension screen for additional space. When you return from the extension screen, the plus sign (+) character appears in the one-byte field.

Show/Purge screen

You can access the Show/Purge Screen Definition (S114) screen from the Online Program Definition menu by entering:

- **SH** or **PU** in the FUNCTION field
- **SD** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field

Alternately, on the List Panel Definitions (P401) screen, you can enter **S** or **P** as a line command for a screen definition (SD).

Field Definitions

The Show/Purge Screen Definition screen fields are the same as the Create/Update Screen Definition screen fields.

COMMAND

For information, see Primary Commands.

You can also enter one of the following commands to invoke the custom code editor for a specified member:

- CREATE *member-name* [*member-description*]
- EDIT *member-name* [*member-description*]
- UPDATE *member-name*
- SHOW *member-name*

OPTIONS

Other TDF functions to complete the necessary specifications of the program definition. Enter any single non-blank character in the input field to the right:

Option You Can Select	Resulting Screen Display
CUSTOM CODE	List/Show Custom Code
DATA GROUP	Create/Update Data Group
PANEL DEF	Update Panel Fields (Online)
ENV <i>environment</i>	The appropriate update screen environment screen, as specified on the Update Session Controls screen
SCRN PARMS	Update/Show Screen Parameters
STORED PROCEDURES	List Stored Procedures to be called

DESC

Description entered on the Online Program Definition menu. You can modify the description here.

(Edit Flag fields) A

Some fields on this screen that you supply custom code member names are preceded by a one-position edit option field. The following list provides valid edit option values and the functions that they invoke:

U

CA Telon transfers control to a blank edit screen allowing you to create a custom code member. If you have already created a custom code member, CA Telon transfers to the List Custom Code screen after you enter the custom code member name in the associated field.

If you have not specified a custom code member in the associated field, CA Telon automatically creates a custom code member and gives it the name of the corresponding entry point. For example, if you enter **U** in the edit option field for OINIT1 but have not specified a name, CA Telon names the custom code member OINIT1. The next time you access the Create/Update Screen Definition screen, the value ****DFLT**** is displayed in the name field to signify that the name matches the entry point name and that the field is protected.

O

CA Telon erases the value displayed in the associated field, including the value ****DFLT****.

This action simply eliminates the association between this entry and the custom code member. It does not delete the member. Its purpose is to allow you to rename the custom code member or associate it with another entry point.

S

CA Telon passes control to the custom code editor and returns the requested member in show mode.

REMARKS

Name of the custom code member to add to the COBOL REMARKS section of the program or to the beginning of the PL/I program.

NEXTPGM

ID of the next program in the application to receive control from the current program. The actual length of the NEXTPGM field is set at CA Telon installation.

Note: If the program transfer control is handled using select field options, PF-key coding, or consistency edit coding, do not enter a value in this field. If you specify a value, CA Telon passes control to the program identified, unless PF-key action specifies the next program and sets the PF-key return code field (CONTROL-INDICATOR) to R.

CURSOR

Label name of the field where the cursor is positioned when the application user initially accesses the screen. Alternately, you can position the cursor through custom code by using the CURSCUS field.

Note: The label name can be the label of only an input, outin, or select field.

SIZE

Number of lines and columns on the screen, in the format // ccc:

//

Two-digit number indicating the number of lines

ccc

Three-digit number indicating the number of columns

SIZE defaults to the size of the panel definition, which is either 24 x 80 or the value specified for the PANEL SIZE field on the Update Sessions Controls screen. If overridden, the size must be larger than the size of the associated panel definition.

LANG

Programming language in which CA Telon is to generate the program. This field is not used in installations using only one language. Values are:

COB

COBOL/LE, COBOL II

PLI

PL/I

The language specified here overrides the default language set on the Update Program Definition Defaults screen. If the default was not set, a value in this field is required.

CMPLOPT

Compiler parameters to be included in the generated program before the COBOL IDENTIFICATION DIVISION line or the PL/I PROC statement. The field on this screen contains 16 bytes. Once an entry has been made in the field, an extension field is presented after the field. If you need to enter a longer value, place a "U" in the extension field to go to the "Update Parameter Overflow" screen, where you can enter a total of 253 bytes, including commas.

IDENTIF

Custom code COPY member name to be added after the COBOL IDENTIFICATION DIVISION line for specification of INITIAL and other Identification Division options, or in the parentheses after OPTIONS in the PL/I PROC statement.

Note: When this copybook is used for PL/I, "MAIN" must be coded if it is desired.

PROCEDR

Custom code COPY member name to be added before the PROCEDURE DIVISION line for specification of Declaratives after the Procedure Division. This copybook is valid only for COBOL; it does not appear on the screen for a PL/I program.

XFERWKA

Custom code COPY member names to add to the TRANSFER WORK AREA section of the program.

You can specify a maximum of 20 COPY member names, each separated by a comma, and a maximum string length of 253 bytes.

You must enter a value for this field, unless you are performing a create function and you specified a XFERWKA list on the Update Program Definition Defaults screen. In this case, that value is the default for this field.

WKAREA

The names of the COPY member(s) that contain a definition of a work area to add to the DATA DIVISION section of the COBOL program. The COPY members can be included as part of the screen definition or as members of a library.

You can specify a maximum of 20 COPY members, each separated by a comma, and a maximum string length of 253 bytes.

Note: COPY members that you specify here are in addition to the standard application COPY member named *hhWKAREA*, where *hh* is the variable application header.

See the *Programming Concepts Guide* for information on *hhWKAREA*.

OINIT1

Name of the custom code member to place in the A-100-OUTPUT-INIT section *before* the automatic database or file read statements (that is, to perform special I/O, initialize areas, and so on).

If there is no automatic read, CA Telon still inserts this custom code member in the same section or procedure and can use the custom code member for custom database or file reads.

OINIT2

Name of the custom code member to place in the A-100-OUTPUT-INIT *after* automatic database or file read statements.

If there is no automatic read, CA Telon still inserts this custom code member in the same section or procedure and can use the custom code member for custom database or file reads.

CURSCUS

Name of the custom code member that contains cursor positioning logic. See the *Programming Concepts Guide* for more information on cursor positioning.

Note: If you are using the CURSOR field, CA Telon positions CURSCUS custom code after the code generated to set the cursor.

OUTTERM

Custom code added at the end of the B-100-OUTPUT-EDITS section. This logic is performed after output edit processing and after SEGLOOP processing (if defined).

PFKEYS

COPY or INCLUDE code to process PF keys to include in the P-100-PFKEYS section of the generated program. CA Telon generates a COPY or INCLUDE statement for each entry.

As installed, the format is *hhPFKnnn*, where:

hh

Value in the Online Program Definition menu HEADER field

nnn

PF key number. The value is generally 1 to 24 but can be any one- to three-character value; for example, the value OTH means other PF keys.

Note: The scroll keys (page forward and page backward) established at installation are automatically requested when paging is specified on the Create/Update File Segloop screen.

The maximum string length is 253 bytes.

ININIT1

Name of the custom code added in the D-100-INPUT-INIT section of the program, prior to any auto exec.

ININIT2

Name of the custom code added in the D-100-INPUT-INIT section of the program, following any auto exec.

SELECT FIELDS

Enter any non-blank character in this field to access the Update Select Fields screen. The TDF presents the Update Select Fields screen only if the panel has select fields.

Note: This field does not show presence or absence of consistency edits or select fields. This information is given on the Update Select Parameters screen.

FLDEDIT

Name of the custom code added in the E-100-INPUT-EDITS section of the program. CA Telon executes this custom code after performing all individual field edits.

SCREEN XFEDIT/ SEGEDIT

Enter any single non-blank character in this field to access the List Consistency Edits screen. If a consistency edit exists, this field contains a plus sign (+).

CONSIS

Name of the custom code added in the X-100-CONSIS-EDITS (consistency edits) section of the program.

INTERM

Name of the custom code added in the H-100-INPUT-TERM (termination) section of the program, following any auto exec.

SECTION

Name or names of the custom code member or members that contain the COBOL sections or PL/I procedures.

You can perform the section or procedure from other parts of the program. You can store the section in a common library, or the section can be unique to this screen.

The maximum number of COPY members you can include is 35, each separated by a comma, and the maximum input value for this field is 253 bytes including commas.

PGMCUST

Name of the COBOL section or PL/I procedure in which to add custom code, and the name of the custom code member to be added. The default is the value in the PGMCUST field on the Update Program Definition Defaults screen. You can make multiple specifications using this format:

section-name1, member-name1,
section-name2, member-name2,...

Section-name

Four-character identifier of the section or procedure in which to include the custom code (for example, H100) and a suffix (I or T) that specifies whether to include the code at the beginning (I) of the section or procedure, or at the end (T).

For example, H100I specifies section H100 is included at the beginning of the program and E100T specifies section E100 is included at the end of the program.

Member-name

Name of the custom code added at the location specified by *section-name*.

Thus, the value A100I,OUTIDC specifies the custom code named OUTIDC is placed at the beginning of the A-100 section.

Note: PGMCUST must be specified in the SETENV field of MACLIB member TLNIIS for any program to use an exit name.

You can include PGMCUST at the beginning and end of any section or procedure within the program, with the exception of U-100 sections or procedures generated by I/O statements (that is, CREATE, DELETE, READ, and UPDATE). On a CREATE statement, the value defaults to profile PGMCUST field.

The maximum input value for this field is 253 bytes including commas.

The following Section names are available for Screen programs:

A100I

A-100-OUTPUT-INIT (beginning of section)

A100T

A-100-OUTPUT-INIT (end of section)

B100I

B-100-OUTPUT-EDITS (beginning of section)

B100T

B-100-OUTPUT-EDITS (end of section)

C100I

C-100-TERMIO-READ (beginning of section)

C100T

C-100-TERMIO-READ (end of section)

C200I

C-200-TERMIO-WRITE (beginning of section)

C200T

C-200-TERMIO-WRITE (end of section)

C210I

C-210-TERMIO-WRITE-INITIAL (beginning of section)

C210T

C-210-TERMIO-WRITE-INITIAL (end of section)

C220I

C-210-TERMIO-WRITE-SUBQUENT (beginning of section)

C220T

C-210-TERMIO-WRITE-SUBQUENT (end of section)

C300I

C-300-TERMIO-XFER (beginning of section)

C300T

C-300-TERMIO-XFER (end of section)

C400I

C-400-TERMIO-MSG-SWITCH (beginning of section)

C400T

C-400-TERMIO-MSG-SWITCH (end of section)

C500I

C-500-FORM-INIT (beginning of section; character client)

C500T

C-500-FORM-INIT (end of section; character client)

C600I

C-600-PROCESS-FORM (beginning of section; character client)

C600T

C-600-PROCESS-FORM (end of section; character client)

C700I

C-700-TP-TO-CLI-ATTR (beginning of section; character client)

C700T

C-700-TP-TO-CLI-ATTR (end of section; character client)

C710I

C-710-TP-SEARCH-TABLE (beginning of section; character client)

C710T

C-710-TP-SEARCH-TABLE (end of section; character client)

C730I

C-730-ATTRIB-LOOP (beginning of section; character client)

C730T

C-730-ATTRIB-LOOP (end of section; character client)

C800I

C-800-CLI-TO-TP-ATTR (beginning of section; character client)

C800T

C-800-CLI-TO-TP-ATTR (end of section; character client)

C810I

C-810-CLI-SEARCH-TABLE (beginning of section; character client)

C810T

C-810-CLI-SEARCH-TABLE (end of section; character client)

C830I

C-830-ATTRIB-LOOP (beginning of section; character client)

C830T

C-830-ATTRIB-LOOP (end of section; character client)

C910I

C-910-GET-MESSAGE & C-910-TERMIO-SAVE (beginning of section; IMS screen programs and drivers)

C910T

C-910-GET-MESSAGE & C-910-TERMIO-SAVE (end of section; IMS screen programs and drivers)

C920I

C-920-GET-WORKSPA (beginning of section; IMS screen programs and drivers with WORKSPA)

C920T

C-920-GET-WORKSPA (end of section; IMS screen programs and drivers with WORKSPA)

C925I

C-925-INSERT-WORKSPA (beginning of section; IMS screen programs and drivers with WORKSPA)

C925T

C-925-INSERT-WORKSPA (end of section; IMS screen programs and driver with WORKSPA)

C930I

C-930-INPUT-MERGE (beginning of section)

C930T

C-930-INPUT-MERGE (end of section)

C935I

C-935-INPUT-MERGE-LOOP (beginning of section)

C935T

C-935-INPUT-MERGE-LOOP (end of section)

C940I

C-940-OUTPUT-MERGE (beginning of section)

C940T

C-940-OUTPUT-MERGE (end of section)

C945I

C-945-OUTPUT-MERGE-LOOP (beginning of section)

C945T

C-945-OUTPUT-MERGE-LOOP (end of section)

C948I

C-948-OUTPUT-OUTIFIL (beginning of section)

C948T

C-948-OUTPUT-OUTIFIL (end of section)

C950I

C-950-PUT-WORKSPA (beginning of section; IMS screen programs and drivers with WORKSPA)

C950T

C-950-PUT-WORKSPA (end of section; IMS screen programs and drivers with WORKSPA)

C960I

C-960-PUT-SPA (beginning of section; IMS screen programs and drivers with SPA)

C960T

C-960-PUT-SPA (end of section; IMS screen programs and drivers with ORKSPA)

C970I

C-970-PUT-MESSAGE (beginning of section)

C970T

C-970-PUT-MESSAGE (end of section)

C990I

C-990-BUFFER-INIT (beginning of section)

C990T

C-990-BUFFER-INIT (beginning of section)

C995I

C-995-BUFFER-INIT-LOOP (beginning of section)

C995T

C-995-BUFFER-INIT-LOOP (end of section)

C999I

C-999-TERMIO-EXEC (beginning of section)

C999T

C-999-TERMIO-EXEC (end of section)

D100I

D-100-INPUT-INIT (beginning of section)

D100T

D-100-INPUT-INIT (end of section)

E100I

E-100-INPUT-EDITS (beginning of section)

E100T

E-100-INPUT-EDITS (end of section)

E200I

E-200-PROCESS-FIELD (beginning of section)

E200T

E-200-PROCESS-FIELD (end of section)

H100I

H-100-INPUT-TERM (beginning of section)

H100T

H-100-INPUT-TERM (end of section)

J100I

J-100-SELECT (beginning of section)

J100T

J-100-SELECT (end of section)

K100I

K-100-HOLD-RESTORE (beginning of section)

K100T

K-100-HOLD-RESTORE (end of section)

K200I

K-200-HOLD-RESUME (beginning of section)

K200T

K-200-HOLD-RESUME (end of section)

K300I

K-300-WINDOWS (beginning of section)

K300T

K-300-WINDOWS (end of section)

K310I

K-310-WINDOW-SAVE (beginning of section)

K310T

K-310-WINDOW-SAVE (end of section)

K320I

K-320-WINDOW-CLEANUP (beginning of section)

K320T

K-320-WINDOW-CLEANUP (end of section)

K330I

K-330-WINDOW-RESTORE (beginning of section)

K330T

K-330-WINDOW-RESTORE (end of section)

L100I

L-100-HOLD-SAVE (beginning of section)

L100T

L-100-HOLD-SAVE (end of section)

L990I

L-990-INSERT-HOLD (beginning of section)

L990T

L-990-INSERT-HOLD (end of section)

M100I

M-100-HELP-ANALYZE (beginning of section)

M100T

M-100-HELP-ANALYZE (end of section)

MAINFIELD

MAIN-FIELD-PROCESS (end of section; CICS character client/server)

MAINFIELDI

MAIN-FIELD-PROCESS (beginning of section; CICS character client/server)

MAINI

MAIN (beginning of section)

MAINT

MAIN (end of section)

MAININPUTI

MAIN-INPUT (beginning of section)

MAININPUTT

MAIN-INPUT (end of section)

MAINLINE

MAIN (replaces entire section)

MAINOUTPUTI

MAIN-OUTPUT (beginning of section)

MAINOUTPUTT

MAIN-OUTPUT (end of section)

MAINPROCESSI

MAIN-PROCESS (beginning of section)

MAINPROCESST

MAIN-PROCESS (end of section)

MAINTERMI

MAIN-FORM-TERM (beginning of section; CICS character client)

MAINTERMT

MAIN-FORM-TERM (end of section; CICS character client only)

N100I

N-100-CURSOR-POSITION (beginning of section)

N100T

N-100-CURSOR-POSITION (end of section)

P100I

P-100-PFKEYS (beginning of section)

P100T

P-100-PFKEYS (end of section)

Q100I

Q-100-CICS-INIT (beginning of section)

Q100T

Q-100-CICS-INIT (end of section)

Q200I

Q-200-PSB-SCHEDULE (beginning of section)

Q200T

Q-200-PSB-SCHEDULE (end of section)

Q210I

Q-210-PSB-TERM (beginning of section)

Q210T

Q-210-PSB-TERM (end of section)

Q300I

Q-300-ACQUIRE-WORKAREAS (beginning of section)

Q300T

Q-300-ACQUIRE-WORKAREAS (end of section)

Q400I

Q-400-ACQUIRE-SPA (beginning of section)

Q400T

Q-400-ACQUIRE-SPA (end of section)

T100I

T-100-CLOSE-FILES (beginning of section)

T100T

T-100-CLOSE-FILES (end of section)

X100I

X-100-CONSIS-EDITS (beginning of section)

X100T

X-100-CONSIS-EDITS (end of section)

Z100I

Z-100-SECTIONS-COPY (beginning of section)

Z900I

Z-900-SECTION-FALLOUT & Z-900-PROGRAM-END (beginning of section);
COOL programs only)

Z970I

Z-970-IDMSSQL-ERROR (beginning of section)

Z970T

Z-970-IDMSSQL-ERROR (end of section)

Z980I

Z-980-ABNORMAL-TERMINATION (beginning of section)

Z980T

Z-980-ABNORMAL-TERMINATION (end of section)

Z990I

Z-990-PROGRAM-ERROR (beginning of section)

Z990T

Z-990-PROGRAM-ERROR (end of section)

Create/Update Data Group

Access

On the Online Program Definition menu, the Nonterminal Program Definition menu, or the Batch Program Definition menu, enter:

- **CR** or **UP** in the FUNCTION field
- **DG** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field (required for function CR)

Program ID

S125

Function

Specifies:

- The data items (DL/I or SQL databases, VSAM or sequential files, and CICS queues or journals) to access for this program definition
- Auto exec and user exec I/O requests

UPDATE DATA GROUP ---- XXXXX.SD				SIZE _____	COL ____
COMMAND ==>				SCROLL ==>	----
	LABEL	REQUEST	KEY/WHERE	IGNORE	
-----	-----	-----	-----	-----	
A	-----	-----	-----	-----	

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

You can also enter the DGADD (Data Group Add) command in the COMMAND field. DGADD requires a parameter (for example, DGADD *psb-name*). This table identifies the possible DGADD parameters and their functions:

DGADD Parameter	Comments
<i>screen-id</i> .SD	Add the data access group from <i>screen-id</i> to the current screen's data group.
<i>psb-name</i> .PSB <i>psb-name</i> .PSB	Add <i>psb-name</i> to the current screen's data group. <i>Psb-name</i> must be a PSB defined in data administration.
<i>filegroup</i> <i>filegroup</i> .FG	<i>Filegroup</i> must be defined in data administration.
<i>hhiii</i> .SD <i>hhiii</i> .BD	<i>Hhiii</i> is the header and ID of a previously defined screen or batch program.
<i>dbd-name</i> .DBD <i>dbd-name</i> .DB <i>dbd-name</i> .DL <i>dbd-name</i> .DLI	<i>Dbd-name</i> is the name of a DBD defined in data administration.
<i>tln-name</i> .DB2 <i>tln-name</i> .D2 <i>tln-name</i> .TABLE <i>tln-name</i> .TAB <i>tln-name</i> .TB	<i>Tln-name</i> is the name assigned to an SQL table defined in data administration.
<i>tln-name</i> .JOIN <i>tln-name</i> .DJ <i>tln-name</i> .TJ	<i>Tln-name</i> is the name assigned to an SQL join defined in data administration.
<i>ddname</i> .VSAM <i>ddname</i> .VSM	<i>Ddname</i> is the name assigned to a VSAM data set defined in data administration. If not defined there, skeletal version of the data set is added to the data group.

DGADD Parameter	Comments
<i>ddname.CQ</i> <i>ddname.CQTS</i> <i>ddname.CQTD</i> <i>ddname.TD</i> <i>ddname.TS</i> <i>ddname.CJ</i>	<p><i>Ddname</i> is the name assigned to a CICS temporary storage or transient data queue or journal defined in data administration. If this queue or journal is not defined in data administration, a skeletal version of the appropriate item is added to the data group. The name qualifier indicates the type of queue:</p> <p>CQ Temporary storage queue</p> <p>CQTS Temporary storage queue</p> <p>CQTD Transient data queue</p> <p>TD Transient data queue</p> <p>TS Temporary storage queue</p> <p>CJ Journal</p>
<i>ddname.SQ</i> <i>ddname.SEQ</i>	<p><i>Ddname</i> is the name assigned to a sequential file defined in data administration. If this data set is not defined in data administration; a skeletal version of the data set is added to the data group.</p>
IMS Only <i>TPPCB-name.TP</i> <i>TPPCB-name.TPPCB</i>	<p><i>TPPCB-name</i> is the name assigned to a TP PCB added to the data</p>
<i>queue-name.CQTS</i> <i>queue-name.TS</i>	<p>Add a temporary storage queue; <i>Queue-name</i> can be up to eight characters long.</p>
<i>queue-name.TD</i> <i>queue-name.CQTD</i>	<p>Add a transient data queue. <i>Queue-name</i> is limited to four characters.</p>
<i>queue-name.CQ</i>	<p>Add a transient data or temporary storage queue defined in data administration. If there are no TD or TS queues defined in data administration, the name defaults to the TS queue.</p>
<i>journal-name.CJ</i>	<p>Add a CICS journal. <i>Journal-name</i> can be up to eight characters long. Note that this is a name that identifies the journal to CA Telon but has no significance to the CICS system in which it is used.</p>

If you specify a file group name, CA Telon appends the file group to the end of the current data group. If you specify a screen definition name, CA Telon appends the data group from that screen to the end of the current data group.

Adding a segment to a DBD

To add a segment to a DBD that you already added to a program on the TDF, follow these steps:

1. Make sure the new segment has been added to the DBD on the Create/Update DBD screen
2. Issue the DGADD command for the DBD so that CA Telon appends the DBD to the program's current data group
3. From the new lines that appear, delete the PCB line (it should be a duplicate) and SEG lines for any segments that already exist in the data group
4. If the new segments are not directly under the DBD with which they are associated, move them by using line commands

Note: CA Telon removes the DG separator line dividing new segments from the previously existing DBD when you leave this screen.

5. Edit and add I/O lines for the new segments, as needed.

(Line command) A

You can modify the screen contents by using these control characters:

Command	Description
C	Copy a line.
CC	Copy a block of lines.
I	Insert a line
<i>Inn</i>	Insert <i>nn</i> lines.
<i>IS(n)</i>	Insert <i>n</i> lines for you to enter the command.
M	Move a line.
MM	Move a block of lines.
R	Repeat a line.
RR	Repeat a block of lines.
A	Line(s) to insert, copy, or move go after this line.
<i>AM(n)</i>	Line(s) to copy or move are repeated <i>n</i> times after this line.
B	Line(s) to insert, copy, or move go before this line.
<i>BM(n)</i>	Line(s) to copy or move are repeated <i>n</i> times before this line.

Command	Description
D	Delete an entry.
F(<i>n</i>)	Redisplay <i>n</i> lines that were excluded with the X or XX command, beginning at the first excluded line.
L(<i>n</i>)	Redisplay <i>n</i> lines that were excluded with the X or XX command, beginning at the last excluded line.
U	Update detail for a data access level.
V	Preview function.
X(<i>n</i>)	Exclude <i>n</i> lines beginning with this line. ¹
XX	Exclude a block of lines. ¹

Note: ¹ - When one or more lines are excluded from the Data Group and the Create/Update Data Group screen is saved on exit, the exclusion is retained. The next time the Data Group is displayed with the exclusion, issue the "RESET" command to remove the exclusion.

See Line Commands for more information.

LABEL

On lines other than I/O edit lines, the COBOL or PL/I data item to which the following access refers.

On user I/O lines, LABEL is an optional field. You can identify the user exec when you require more than one of the same type of I/O. CA Telon uses the value in this field to generate the name of the COBOL or PL/I paragraph or procedure that contains the I/O.

- OUTREAD, OIREAD, and INREAD access types— LABEL must be AUTOEXEC because CA Telon generates and performs these I/O requests automatically.
- BROWSE, CREATE, and UPDATE access types— You have the option of using the AUTOEXEC label to have CA Telon generate and perform the associated I/O. If you do not specify a label, CA Telon identifies the requests as USEREXEC I/O, and CA Telon generates but does not perform the I/O. You must write your own code to perform user exec requests.

Note: AUTOEXEC is a CA Telon reserved word for LABEL meaning automatically generated and executed CA Telon code.

The first character of a value in this field must be a letter, @, #, or \$. Subsequent characters may be numbers. No other special characters are allowed. Use of characters other than these will result in Generator errors.

Note: CA Telon automatically performs BROWSE even if you do not specify the label AUTOEXEC (for SQL, if you specify AUTOEXEC, CA Telon performs I/O from B-100 rather than generating I/O in-line).

Example:

		LABEL	REQUEST	...
Segment	==>	TRGEMPL	DEFINE	
user exec	==>	READ	READNEXT...	
		READNEXT	READNEXT...	

This generates these user exec paragraphs:

- In COBOL, U-100-READ-TRGEMPL and U-100-READNEXT-TRGEMPL
- In PL/I, U_100_READ_TRGEMPL and U_100_READNEXT_TRGEMPL

REQUEST

On lines other than I/O edit lines, information that you can use to identify the data item to which the following I/O applies. For example, on a join, CA Telon displays a marker for a join. For a SEG, ROW, or REC line, CA Telon displays the usage of the segment, row, or record. REC is used when defining queues, journals, and VSAM and sequential files.

The data group editor maintains usages of @DEFINE and @DUMMY, depending on the existence of I/O for the SEG, ROW, or REC. You can update usage by updating the appropriate SEG, ROW, or REC edit line. Change the usage from @DUMMY to DEFINE so that CA Telon will set up I/O areas and related storage areas without generating any data access for the SEG, ROW, or REC in question.

Usage

Values for usage are:

Request	Action
BROWSE	Read the SEG/ROW in a loop during the program output processing.
DEFINE	Explicitly request CA Telon to generate storage areas for this REC/SEG/ROW, regardless of whether data access has been defined for this item.
@DEFINE	CA Telon assumes that this item needs storage areas because this REC/SEG/ROW has data access defined to it. If you remove all data access for this REC/SEG/ROW, this automatically reverts to @DUMMY.

Request	Action
DUMMY	The developer has explicitly requested that CA Telon should not generate storage areas for this REC/SEG/ROW, regardless of whether data access has been defined for this item.
@DUMMY	CA Telon assumes that the REC/SEG/ROW is not being used and no storage areas are needed. This is switched to @DEFINE when you insert the type of data access used.
HOLD	Used for HOLD processing.
WORKSPA	CA Telon handles the SEG/ROW as a workspa.

Data access lines

For data access lines, you can enter any of the following values for this field:

Request	Action
BROWSE	Used with AUTOEXEC label only (in conjunction with SEGLOOP) for READ processing in a loop during output processing.
CREATE	Can be used with AUTOEXEC label. Insert or write during input processing.
DEFINE	The segment is defined to the TDF. CA Telon processes it in the generation procedure to create storage areas for it.
DELETE	Can be used with AUTOEXEC label. Delete during input processing.
HOLD	Used for CA Telon hold processing.
INREAD	Used with AUTOEXEC label only. Read during input processing initialization.
JOURNAL	Can be used with the AUTOEXEC label. It writes a record to the CICS journal.
MATCHM	<p>In a match-structure program, you must select files for auto exec match handling. Select a file as the match master file and assign an auto exec MATCHM request to it. Select another file as the match transaction file and assign an auto exec MATCHT to it.</p> <p>Standard line commands are valid for these requests. The U command displays the update detail data access screens. PREVIEW is not supported for MATCHM and MATCHT auto exec requests.</p>

Request	Action
MATCHT	See MATCHM above.
MERGE nn	<p>Select up to 20 files for auto exec merge handling in a merge-structure program. The files are identified hierarchically by the requests MERGE01 through MERGE20. This means the file with request MERGE01 is accessed first, MERGE02 second, and so on. In assigning the MERGEnn data access requests, you must increment nn by 1. For example, four requests must be identified MERGE01, MERGE02, MERGE03, MERGE04.</p> <p>Standard line commands are valid for these requests. The U command displays the update detail data access screens. PREVIEW is not supported for MERGEnn auto exec requests.</p>
OIREAD	Used with AUTOEXEC label only. Read during output and input initialization processing.
OUTREAD	Used with AUTOEXEC label only. Read during output initialization processing.
READ	<p>For DL/I, READ is used to get a unique segment unless you specify a function. Then it performs a different type of get depending upon the function.</p> <p>For VSAM and SEQ, READ is used to perform a read.</p> <p>For SQL, READ is used to perform a select.</p> <p>For CICS TS and TD Queues, READ is used to perform a READQ.</p>
READNEXT	<p>For DL/I, READNEXT is used for a get next.</p> <p>For SQL, READNEXT is used to open a cursor and perform a fetch from the cursor.</p> <p>For VSAM and QUEUES, READNEXT is used to perform a read next.</p> <p>For SEQ, READNEXT is used to perform a read.</p> <p>For CICS TS and TD Queues, READ is used to perform a READQ.</p>
REPLACE	<p>For DL/I, REPLACE is used for a REPLACE.</p> <p>For SQL, REPLACE is used to perform an UPDATE.</p> <p>For VSAM, REPLACE is used to perform a REWRITE.</p>
SPBROWSE	Used with AUTOEXEC label only (in conjunction with SEGLOOP) for READ processing in a loop during output processing on a program that calls a stored procedure which returns a result set.
SPRDNEXT	SPRDNEXT is used by a program that calls stored procedure to perform a fetch from a cursor that has been opened by the

Request	Action
	called stored procedure.
SPTRNACT	Used with AUTOEXEC only (in batch and CICS nonterminal programs which call stored procedures). A fetch from a cursor that has been opened by the called stored procedure is generated in the GET-TRAN section.
TRANSACT	Used with AUTOEXEC only (in batch and CICS nonterminal programs only). Sequential read is generated in the GET-TRAN section.
UPDATE	Can be used with AUTOEXEC label. With AUTOEXEC, CA Telon generates a read during output initialization. Then CA Telon generates a read for update followed by an update during input processing. When AUTOEXEC is not specified, CA Telon generates but does not perform the read for update followed by an update only. Note: All data access requests that generate READs are invalid for CICS journals. CREATE, DEFINE, @DEFINE, DUMMY, and @DUMMY are the only valid REQUEST values for CICS journals.
WORKSPA	CA Telon handles the segment as a workspa.

Queue and journal auto exec I/O requests

Data Type	Item	Auto exec request
Queues	CQUE	OUTREAD, INREAD, OIREAD, CREATE; UPDATE, BROWSE (TS only)
Journals	CJNL	JOURNAL

Queue and journal user exec I/O requests

Data Type	Item	Auto exec request
Queues	CQUE	READ, CREATE, DELETE, READNEXT; REPLACE, UPDATE (type TS only)
Journals	CJNL	JOURNAL

KEY/WHERE

This field can contain:

- The host variable name(s) containing the key(s) that identify the segment or record accessed by the I/O
- The entire WHERE clause

CA Telon uses the value in this field:

- To set the key in the SSA, in DL/I
- As the RIDFLD, in VSAM
- To construct the WHERE clause, in SQL and EXEC DLI

IGNORE

Status codes to be ignored. All DBMS-specific status codes are valid. Generic status values are translated to the corresponding DBMS-specific status codes at generation time.

This table presents generic values and how CA Telon handles generic values:

Generic Status	What Is Ignored
OK	Blank or zero status codes
NOTFOUND or NFD	Not found status codes
DUPLICATE or DUP	Multiple record or key occurrences
LOGICERR or LOG	Errors on access that depends on prior required conditions
SECURITY or SEC	Security violations
ENDFILE or EOF	EOF conditions
NOTAVAIL or NAV	Conditions when resources are not available
DBMERROR or DBM	Any return code not specified above (SQL only)
REF	For DB2, referential integrity errors -530 through -536
ALL	Any return code

If CA Telon encounters a return code other than blanks that is not included in the IGNORE field for an I/O, CA Telon assumes ABEND processing as follows:

- For all I/O except BROWSE, CA Telon invokes ABEND processing. OK is the default.
- For BROWSE I/O, CA Telon does not invoke ABEND processing. ALL is the default.

Update DBMS Characteristics

Access

On the Create/Update Data Group screen, enter **U** as a line command for any DBMS type.

Program ID

S127

Function

Changes the DBMS type of the COBOL or PL/I data item.

XXXXXX.XX UPDATE DBMS CHARACTERISTICS *****
COMMAND ==> _____

CURR DBMS: TYPE _____
* TLNNAME _____
* PCB _____

SQL ONLY: TABLE _____ (QUAL.TBLNAME)
* SYNONYM __ (Y/N)

DL/I ONLY: PCBNAME _____
* KEYLTH _____
* PROCSEQ _____
* PROCOPT _____

TP ONLY: LTERM _____
* EXPRESS _ (Y/N)
* MSGCALL _ (Y/N)
* ABCALL _ (Y/N)
PRINT _ (Y/N)

Field Definitions

COMMAND

For information, see Primary Commands.

TYPE

(*Protected field.*) Identifies the type of DBMS (for example, SQL). The displayed value is from the Create or Update PSB, File Group screen.

Note: The value REF implies that this database is used for inheritance processing only.

TLNNAME

Type of DBMS:

- For DL/I, the name of the database as specified in the DBD
- For SQL, the label of the table
- For VSAM, the name of the data set

The displayed value is from the Create/Update PSB Or File Group screen.

PCB

(*Protected field.*) Identifies the DL/I PCB name.

The displayed value is from the Create/Update PSB Or File Group screen.

TABLE

The fully-qualified name of the SQL table.

SYNONYM

Option for generating the qualifier for the SQL table. Values are:

Y

CA Telon leaves the qualifier for the table blank. This allows for full qualification by SQL during SQL precompilation.

N

CA Telon generates the qualifier for the table, which is the generated program that contains the fully-qualified SQL table or view names.

PCBNAME

The name of the COBOL or PL/I data item used to access the PCB.

KEYLTH

For a database or a data set, the length of the concatenated key for the PSB that CA Telon generates.

For a teleprocessing PCB, do not enter a value for this field.

PROCSEQ

The name of the secondary index used to process the database named in the PCB statement's DBDNAME operand. The value in this field defines the PCB statement, PROCSEQ=.

PROCOPT

The DB processing options. This field is valid only when DB or GSAM are values in the TYPE field. The value in the PROCOPT field defines the PCB statement, PROCOPT=.

For more information about the DB processing options, see IMS utilities documentation.

LTERM

The destination for messages sent using this PSB. Options include any IMS transaction code and any logical terminal name defined to IMS. The value in this field defines the PCB statement, LTERM=.

If you do not specify a value in this field, the destination for the PCB is modifiable in the program.

EXPRESS

A value to specify whether the PCB processes a message if an application ABENDs. EXPRESS is valid only for teleprocessing PCBs. Values are:

Y

Messages from the alternate PCB are sent if an ABEND occurs

N

Messages from the alternate PCB are backed out if an ABEND occurs

The default depends on the value of the ABCALL field: if ABCALL=Y, then EXPRESS defaults to Y; otherwise, EXPRESS defaults to N.

MSGCALL

A value to specify whether this PCB is used in place of XFER-PCB for generated message switch processing. Values are:

Y

This PCB is used in place of XFER-PCB for generated message switch processing

N

This PCB is not used in place of XFER-PCB for generated message switch processing

ABCALL

(Valid only for teleprocessing PCBs.) A value to specify whether the PCB that CA Telon generates replaces CA Telon's XFER-PCB in calls to the CA Telon ABEND handling routine.

Values are:

Y

(*Default.*) The PCB that CA Telon generates replaces the XFER-PCB.

N

CA Telon uses the XFER-PCB

PRINT

A value to indicate whether the REPORT subroutine uses this PCB. If you do not specify a value, then the print subroutine uses the XFER-PCB. This field is valid only for teleprocessing PCBs.

Values are:

Y

(*Default.*) The PCB that CA Telon generates and replaces the XFER-PCB.

N

CA Telon uses the XFER-PCB

Select New Row Name

Access

On the Create/Update Data Group screen, enter **U** as a line command in a ROW field.

Program ID

S137

Function

Specify the usage and key for the TLNROW; optionally identify a TLNROW as a temporary table.

SELECT NEW ROW NAME *****

COMMAND ==> _____PAGE 01

TABLE NAME: _____CURRENT ROW: _____

USAGE: _____TMPQUAL: _____TMPNAME: _____TMPCOMT: _____

* KEY OR

* WHERE _____

* _____

* _____

S TLNROW C-CNT COLUMNS DEFINED

- _____A

- _____A

- _____A

Field Definitions

COMMAND

For information, see Primary Commands.

Note: You may use the **TEMPTBL<qual.tblname>** command to convert the current row to a global temporary table. Tblname is the name you want to give your temporary table, and is required. Qual is the temporary table name qualifier. If you do not supply a qualifier, the default value SESSION is used.

TABLE NAME

(Protected field.) Identifies the qualifier and name of an SQL table to which the current row belongs.

CURRENT ROW NAME

(Protected field.) Identifies a row in an SQL table. You can alter the value in the S field. Data access uses the column definitions for this row.

USAGE

Row usage. Values are:

DEFINE

Data access requests have been created in this program to use the row.

DUMMY

The row is not referenced in this program. The value is not changed even if a data access request is defined for this row.

@DEFINE

This row is referenced in data access requests in this program. CA Telon controls this inheritance value. If all data access requests referencing this row are deleted from this program, the value is changed to @DUMMY.

@DUMMY

CA Telon assumes that the row is not being referenced in any data access requests in this program. The &DUMMY. value is changed to @DEFINE when a data access request is added to the data group for this row.

TMPQUAL

Identifies the qualifier used for the temporary table. This parameter and its label are only displayed if the current row is defined as a temporary table, either on this screen with the TEMPTBL command described earlier, or on the Create/Update SQL Tables/TLNROWS (D141) screen in Data Administration. If no qualifier is specified, the default of SESSION is used.

TMPNAME

Identifies the qualifier used for the temporary table. This parameter and its label are only displayed if the current row is defined as a temporary table, either on this screen with the TEMPTBL command described earlier, or on the Create/Update SQL Tables/TLNROWS (D141) screen in Data Administration.

TMPCOMT

Identifies the temporary table ON COMMIT value. This parameter and its label are only displayed if the current row is defined as a temporary table, either on this screen with the TEMPTBL command described earlier, or on the Create/Update SQL Tables/TLNROWS (D141) screen in Data Administration. Values are:

SAVE

Generate ON COMMIT PRESERVE ROWS in the EXEC SQL DECLARE command for the temporary table.

DELETE

Generate ON COMMIT DELETE ROWS in the EXEC SQL DECLARE command for the temporary table.

DROP

Generate ON COMMIT DROP TABLE in the EXEC SQL DECLARE command for the temporary table.

KEY OR WHERE

CA Telon uses this field to specify mass insert criteria on an SQL CREATE data access request. This parameter can contain:

- The PL/I or COBOL host variable names containing the keys that identify the rows being accessed.
- The entire SQL key clause. CA Telon uses the value in this field to construct the WHERE clause.

S

Allows you to identify the corresponding TLNROW name as the CURRENT ROW NAME. To select a TLNROW, enter any non-blank character in its S field.

This parameter value directs CA Telon which TLNROW to use as the CURRENT ROW NAME. Any non-blank character may be entered to select a TLNROW. Data access requests for the selected TLNROW have access to the columns defined to it.

TLNROW

(Protected field.) Identifies the columns of the specified TLNROW. There may be multiple columns for each row.

Note that the first TLNROW has the same name as the row named in the CURRENT ROW NAME field unless you change the CURRENT ROW NAME with the SELECT override.

C-CNT

(Protected field.) Identifies the number of columns in the TLNROW.

COLUMNS DEFINED

(Protected field.) Identifies the columns in the TLNROW.

(OUTPUT FIELD) A

(Protected field.) A plus sign (+) indicates that there are more columns associated with the TLNROW than will fit in the COLUMNS DEFINED field. You can view all the columns associated with the TLNROW on the Create/Update SQL Tables/TLNROWS (D141) screen.

Update Database Segment

Access

On the Create/Update Data Group screen, enter **U** as a line command on the listing marked SEG=>.

Program ID

S135

Function

Updates detail segment characteristics valid for any type of I/O.

CA Telon exports values from this screen to the Create/Update DBD screen, Create/Update Data Group screen, and Show/Update Data Set Default Data screen.

UPDATE DBD: _____ SEGMENT: _____ *****

COMMAND ==> _____

OPTIONS ==> PCB PARMS

GENERAL

LABEL _____

USAGE _____

KEYPIC _____

* COPY _____

COPYLV1 _____

COPYLBL _____

* KEYLEN _____

** DSCREF

SEGMENT

CMND

IMSKEY

OP

KEY

_____ - A

Field Definitions

DBD

(*Protected field.*) Identifies the DBD name specified on the NAME field of the Create/Update DBD screen.

SEGM

(*Protected field.*) Identifies the segment specified on the NAME field of the Create/Update DBD screen.

COMMAND

For information, see Primary Commands.

PCB PARMS

PCB parameters. To select this field, enter a nonblank value. This transfers you to the Extended Parameter Utility screen, on which you can enter parameters such as PROCOPT and INDICES. See Update S for information on PROCOPT and INDICES.

LABEL

The segment, when you require more than one occurrence of a segment name in a file group (duplicate segment name). This value must uniquely identify the segment in the file group. It is used in generating names such as SSA or U_100 names.

For example, if you specify READ for segment TRGEMPL, CA Telon generates the paragraph U-100-READ-TREGEMPL (for COBOL).

USAGE

The type of processing associated with the SEGMENT. Values are:

BROWSE

Read the segment in a loop on output.

DEFINE

This segment is being used and must be set for SSA overrides (specified in KEYPIC, KEYLEN, CMND, IMSKEY, OP, and KEY) to be recognized during the generation if this segment contains no data access requests, but any of its child segments containing data access requests reference the ****DFLT**** SSA for this segment.

DUMMY

The segment is not being used.

@DEFINE

This segment is being used and is automatically set when the segment contains data access requests at this level.

@DUMMY

This segment contains no data access requests, but any of its child segments containing data access requests reference the ****DFLT**** SSA for this segment.

HOLD (IMS Only)

Used for hold processing.

WORKSPA (IMS Only)

CA Telon handles the SEG/ROW as a workspa.

KEYPIC

The format of the key for the segment. CA Telon uses the value in this field for the VALUE parameter in the SSA.

If the format of the key for the segment is a character format, do not enter a value.

If the format of the key for the segment is packed decimal or binary and the key value being moved to the SSA is of a different format, the key is converted when it is moved to the SSA.

The following is an example of a COBOL entry for the KEYPIC field:

9(5) COMP-3

COPY

The COBOL COPY or PL/I INCLUDE member name that identifies the segment definition. CA Telon uses the contents of the member for the layout of the segment. The member name should be unique for this program.

If you enter **NONE**, CA Telon does not copy a segment definition into the program.

Note: NONE is not allowed for index data sets.

If you do not make an entry in this COPY field, CA Telon uses the segment name to identify the copy library member name with the layout of the segment.

COPYLV1

The start level of the COPY or INCLUDE member. Values are:

Y

The COPY/INCLUDE member for the I/O area of this segment starts at the COBOL or PL/I 01 level. You must also specify a value in the COPYLBL field to supply the I/O area for the data access that CA Telon generates.

N

CA Telon expects the I/O area to begin at the 03 level or higher. In this case, CA Telon generates the 01 and 02 levels and uses the COPY/INCLUDE member for 03 levels and below.

COPYLBL

The COBOL or PL/I data item name of the group level (for example, an 01 field and its subordinates) for the segment copy definition. The COPYLBL value overrides the I/O area that CA Telon would normally provide.

The default I/O area in all database and data set calls is *IOA-segment-name*-SEGMENT.

If two segments have the same segment name (the default value of DBSEG or some other name that you specify on the LABEL field) and the same COPYLBL, CA Telon generates only one I/O area. Thus, in order for CA Telon to generate a second I/O area, enter a unique data item name in this field. If CA Telon is generating any automatic I/O, and you have specified the COPYLV1 field value Y, you must enter a value.

In this field, you can enter either a member name or **NONE**. The member name is the name of the COBOL COPY member or PL/I INCLUDE containing the segment definition. If the COPY or INCLUDE member has the same name as the segment (value in the SEGMENT field), you need not enter a value in this field— CA Telon automatically includes the COPY or INCLUDE member whose name is the same as the segment name.

KEYLEN

The length of the key used to access the database. CA Telon uses this value to define the length of the key in the SSA that it generates.

DSCREF

The data search criterion (DLIDSC) at this segment level. CA Telon uses this protected value to define the length of the key in the SSA that it generates.

To increment USECNT appropriately, enter **U** or **S** to request the List SSAs screen, and **S** to select DLIDSC for this segment level or I/O request. See Error! Reference source not found. for more information on incrementing USECNT.

SEGMENT

(*Protected field.*) Identifies a segment.

CMND

Overrides the SSA command code for the segment that CA Telon generates. The characters you specify are prefixed with an asterisk (*) and suffixed with one to three dashes (-) to make a four-byte code.

For example, if you enter **D**, CA Telon generates the SSA command code as:

*D--

If you omit this field, CA Telon uses the IMS default *---.

Note: For EXEC DLI, you can use this field to specify segment qualification options. EXEC DLI does not support all command codes.

IMSKEY

The name of the key or search field for the segment defined in the DL/I DBD. By default, this value is initially set to the segment's key field or to blank if not keyed. Override this value, if necessary, with any other valid search field defined for this segment.

Note: Specify the KEYPIC and KEYLEN corresponding to the key or search field specified here.

OP

For DL/I processing, the value used to define the relational operator in the SSA. This field overrides the default OP CODE in the SSA that CA Telon generates.

If you do not specify a value, CA Telon uses the OP CODE '>=' for BROWSE and keyed READNEXT data access, and '=' for other keyed access.

Values are:

Value	Description
=	Equal to
>=	Greater than or equal to
=>	Greater than or equal to
<=	Less than or equal to
=<	Less than or equal to
>	Greater than
<	Less than
≠	Not equal to
=≠	Not equal to

Note: OP does not apply for unqualified SSAs; that is, for BROWSE requests for which there is no starting key value (STBRKEY, defined on the Create/Update File Segloop screen). If you use the SCHFLDx field on the Create or Update File SEGLOOP screen, OP can (but need not always) be set to the equal sign (=).

KEY

The PL/I or COBOL variable name containing the key to a segment. If inheritance is requested, data access (user I/O) uses the variable name to identify a segment that it retrieves. See the *Programming Concepts Guide* for information on inheritance.

Note: If USAGE is BROWSE, the KEY field can be used in place of the SEGLOOP field, PAGEKEY. If both are used, KEY is ignored (the use of PAGEKEY is recommended). KEY does not apply for unqualified SSAs.

(KEY extension) A

Enter any non-blank character, with the exception of a plus sign (+) and a question mark (?), for a screen display that allows you to specify a host variable key longer than 30 characters.

Update Data Set Record

Access

On the Create/Update Data Group screen, enter **U** as a line command for a sequential or VSAM record.

Program ID

S136

Function

Updates detail data set segment data.

HEADERID.UPDATE DATA SET RECORD *****

COMMAND ==> _____

DATA SET _____ ACCESS _____ *****

GENERAL LRECL _____ (MIN MAX) BLKSIZE _____

* OPEN _____

RECORD: IGNOPEN __ (Y/N) IGNCLOSE __ (Y/N) IGNEMPT __ (Y/N)

* LABEL _____ USAGE _____

* COPY _____

* COPYLV1 __ (Y/N)

* COPYLBL _____

* COBDIV __ (FD/WS)

* COBVSky _____

I/O: KEY _____

* KEYLEN _____

* OPCODE _____

VSAM: TYPE _____ (KSDS/RRDS/ESDS) ACCMODE __ (DYN/RAN/SEQ)

* OPTLIST _____

* RECLTH _____

* GENKEYL _____ INDEXOF _____

Field Definitions

COMMAND

For information, see Primary Commands.

DATA SET

(*Protected field.*) Identifies the data set to which you are specifying information.

ACCESS

(*Protected field.*) Specifies the type of access (VSAM or SEQ) for the data set that you are defining.

LRECL (VSAM Batch Only)

The length in bytes of the logical record for the data set. For variable-length files, enter the minimum (MIN) and maximum (MAX) lengths of the records.

BLKSIZ (VSAM Batch Only)

The blocking factor, in bytes, of the data set. If not specified here, the blocking factor must be specified in the JCL.

OPEN (VSAM Batch Only)

The way in which CA Telon is to automatically open the data set at program initialization. Values are:

INPUT

Input

OUTPUT

Output

I-O

Input/output

At program termination, CA Telon closes any files automatically opened at program initialization.

IGNOPEN (VSAM Batch only)

Indicates whether or not the dataset's status should be tested after it is opened. Values are:

Y

Test the dataset's OPEN status.

N

Don't test the OPEN status.

IGNCLOS (VSAM Batch only)

Indicates whether or not the dataset's status should be tested after it is closed. Values are:

Y

Test the dataset's CLOSED status.

N

Don't test the CLOSED status.

IGNEMPT: (VSAM Batch only)

Indicates whether or not the dataset's status should be tested when it is opened to see if it is empty. Values are:

Y

Test the dataset's EMPTY status

N

Don't test the EMPTY status

LABEL

The default label associated with this data set. If specified, this value replaces the name of the data set that CA Telon generates.

USAGE

The type of processing associated with the record you are defining. Values are:

BROWSE

Read the record in a loop on output.

DEFINE

The record is defined to the TDF. CA Telon processes it in the generation procedure.

@DEFINE

The record is being used, but CA Telon is controlling it.

DUMMY

The record is not being used.

@DUMMY

CA Telon assumes that the record is not being used. This changes to @DEFINE when you define the type of data access.

HOLD

Used for HOLD processing.

WORKSPA (IMS Only)

CA Telon handles the record as a workspa.

COPY

The COBOL COPY or PL/I INCLUDE member name that contains the record definition. Enter either a member name or **NONE** in this field.

If you enter a member name, CA Telon uses the contents of that member for the layout of the record. Member name must be a unique name for this program.

If you enter **NONE**, CA Telon does not copy a record definition into the program.

COPYLV1

The start level of the COPY or INCLUDE member. Values are:

Y

The COPY/INCLUDE member for the I/O area of this segment starts at the COBOL or PL/I 01 level. You must also specify a value in the COPYLBL field to supply the I/O area for the data access that CA Telon generates.

N

CA Telon expects the I/O area to begin at the 03 level or higher. In this case, CA Telon generates the 01 and 02 levels and uses the COPY/INCLUDE member for 03 levels and below.

COPYLBL

The COBOL or PL/I data item name of the group level (for example, an 01 field and its subordinates) for the record copy definition. The COPYLBL value overrides the I/O area that CA Telon would normally provide.

The default I/O area in all database and data set calls is *IOA-record-name-SEGMENT*.

If two records have the same record name (the default value of DBSEG or some other name that you specify on the LABEL field) and the same COPYLBL, CA Telon generates only one I/O area. Thus, in order for CA Telon to generate a second I/O area, code a unique data item name for COPYLBL. If CA Telon is generating any Automatic I/O, code COPYLBL when you have specified the COPYLV1 value as Y.

For the COPYLBL field, you can specify either a member name or **NONE**. The member name is the name of the COBOL COPY member or PL/I INCLUDE containing the record definition. If the COPY or INCLUDE member has the same name as the record, you do not have to include this field. CA Telon automatically includes the COPY or INCLUDE member whose name is the same as the record name.

If you enter **NONE** for this field, CA Telon does not copy a record definition into the program. NONE is commonly used when reading an input sequential file whose record is defined in a COPY or INCLUDE member in working storage.

COBDIV

The section of the COBOL program in which to copy the record layout of the data set.

The definition is to begin at level 03 or higher and will have 02 IOA-record-name-SEGMENT generated above it.

If not specified, CA Telon copies the record layout into the COBOL file definition (FD) for the data set.

If you code working storage (WS), the record layout appears in the IO-AREA of the program. A value in this field is only valid on batch programs and on a RECORD statement referencing a data set. See the *Programming Concepts Guide* for information on RECORD statements.

COBVSKEY

The data name of the VSAM KSDS or RRDS key for the file.

For a VSAM KSDS, you must define the key for the file by using the KEY field on this screen.

For a VSAM RRDS, the data name must specify an unsigned integer defined in the working storage of the COBOL program. Note that CA Telon does not generate the relative key, so you must define it. COBOL uses this variable keyed access to VSAM files.

A value in this field is valid in only batch programs and on RECORD statements referencing a VSAM key sequenced or relative data set. Enter **DUPLICATE** after the data name to specify the KSDS alternate key (that is, key on a data set using the INDEXOF field) does not have to be a unique data set.

KEY

The PL/I or COBOL variable names containing the key to a record. If inheritance is requested, data access (user I/O) uses the variable name to identify a record that it retrieves. See the *Programming Concepts Guide* for information on inheritance.

KEYLEN

The length of the record identification (RID) field.

OPCODE

The OPCODE that CA Telon determines for VSAM access. Values are:

GTEQ

(*Default.*) Greater than or equal to.

EQ

Equal to

GT

Greater than

TYPE

The type of access for the data set you are defining. Values are:

KSDS

(*Default.*) Key-sequenced data set.

RRDS

Relative record data set

ESDS

Entry-sequenced data set

ACCMODE (VSAM Batch Only)

The access to the type of data set you are defining. Values are:

DYN (COBOL only)

Dynamic access

RAN (COBOL Only)

Random access

SEQ

Sequential access

DIR (PL/I Only)

Direct access

OPTLIST (VSAM Processing Only)

Options on CICS data sets for command level calls. Values are:

- RRN
- SEGSET
- SEGSETALL
- SYSID
- MASSINSERT
- DEBKEY
- DEBREC
- UPDATE

You can specify one or more of these options on the SEGMENT statement and all user exec data access (READ, UPDATE, CREATE, and DELETE). Separate each option with a comma.

Values that you code on the SEGMENT statement are carried down to user exec specifications if you do not code an OPTLIST on the user exec data access.

CA Telon automatically removes invalid values from a call for a particular command level verb. For example, if MASSINSERT is specified as an option on the RECORD statement or user exec data access, it only appears if the EXEC CICS WRITE command is generated for that segment in the CICS program.

You can specify a literal value for the SEGSET and SYSID options by enclosing the literal value in double quotes. For example:

```
OPTLIST=(RRN,SYSID("SYSA"))
```

RECLTH (VSAM processing of variable-length records only)

The maximum length of each record on the file. Values are:

record-length

Maximum record length, specified as either an integer or the name of a COBOL or PL/I variable that contains the key-length value. This value is used when reading or writing the VSAM record. Any rewrite operations are processed using the current length of the record being updated (as determined by the read).

Read-length,rewrite-length

Maximum record length during a read and subsequent update, respectively. Each value can be specified as either an integer or the name of a COBOL or PL/I variable that contains the key-length value. Read-length is used for all automatic read EXEC CICS calls for the record. Rewrite-length is used as the maximum length of the updated record. This specification is applicable for UPDATE processing only (that is, usage is UPDATE).

The value in this field overrides the RECLTH value specified on the RECORD statement for the file being accessed.

Note: The length that you use in a CICS read operation for a variable-length record must be at least as large as the actual record retrieved. If not, a CICS abend results.

GENKEYL (VSAM Processing Only)

The length of the generic key used for the access.

You can specify either an integer or the name of a COBOL or PL/I variable that contains the key-length value. If you do not specify a value, CA Telon assumes that the access uses the full key length; it does not use the GENKEYL value from the RECORD statement.

INDEXOF (VSAM Batch only.)

A data set in this data group of which this data set is an index. The index data set must physically follow the indexed data set on the Create/Update Data Group screen.

Note: The COPY field is not valid for an index.

Update SQL Detail Data Access

Access

On the Create/Update Data Group screen, enter **U** as a line command for a table.

Program ID

S147

Function

Adds SQL-specific information.

Note: The @ sign on this screen indicates that the value is inherited from data administration.

For more information about data inheritance, see the *Programming Concepts Guide*.

HHIII.XX XXXXXXXX XXXXXXX XXXXXXX ** *****

COMMAND ==>

OPTIONS ==> PREVIEW _ GETDIAG _

GENERAL: KEY OR

* WHERE

*

*

* IGNORE

* IOAREA

*

DB2/SQL: SENCOLS

*

*

*

* ORDERBY

*

* GROUPBY

* HAVING

*

* KEYCOLS

*

* OPCODE

CUSTOM CODE: A CPYCALL A CPYINIT RESULTCC A

* A CPYKEY A CPYTERM RESULTPR

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

(Extension Flag field)

You can transfer to the Field Extension screen if you require more space to enter data in these fields:

- KEY or WHERE
- IOAREA
- SENCOLS
- ORDERBY
- GROUPBY
- KEYCOLS

To transfer to the Field Extension screen, enter **U** in the Flag field in column 80 to the right of the field that you need to extend. The Field Extension field redisplay the field with additional lines for data entry.

You can transfer from the Update SQL Detail Data Access (S187) screen to the List Select Columns (S144) screen by entering **S** in the selection field in column 80 to the right of the SENCOLS, ORDERBY, GROUPBY, or KEYCOLS field. This screen allows you to select valid columns from a list.

You can also transfer to the Update User-Defined Datatypes (S184) screen by entering **D** in the selection field.

PREVIEW

Place a non-blank character in this field to display a preview of what the current data access request will look like when generated.

Note: You cannot use PREVIEW to edit or save the previewed data access request. Use the CPYCALL field to edit or save the call. When a CPYCALL copybook exists for this data access request, the CPYCALL custom code is displayed when PREVIEW is selected.

GETDIAG

Enter a non-blank character in this field to transfer to the GET DIAGNOSTICS List (S240) screen.

FTCHOPT

Enter a non-blank character in this field to transfer to the Fetch Details (S244) screen.

KEY OR WHERE

Enter expressions in the following formats:

- The key is a list of host variables, separated by commas and paired with KEYCOLS and OPCODEs to generate the WHERE clause. The default OPCODE is the equals sign (=).

Multiple variables should be separated by commas.

- The key is separated by commas, followed by the OPCODE, followed by the host variables, as shown in the example below:

```
TASK EMPL_ID,=:XFER_EMPL_ID, OR, TASK EPL_ID,=:WS_TASK_ID
```

- The expression appears in one of the following formats:
 - The entire expression is enclosed in single quotes, as shown below:
'COUNT(EMPL_NAME) > 10'
 - The entire expression appears in the following sequence:
 - a. An expression enclosed by single quotes
 - b. A logical operator (OR, AND, or NOT)
 - c. An expression enclosed in single quotes

You can specify the BETWEEN and LIKE predicates using this format only. An example is shown below:

```
:EMPL_NAME BETWEEN "JOHN" AND "JONES"
```

The following considerations apply to KEY or WHERE clauses:

- To generate a literal value, enclose the value in double quotes, as shown below:

```
:EMPL_NAME BETWEEN "JOHN" AND "JONES"
```

Note: Use caution when you specify literals. Because the Generator replaces double quotes (") with two apostrophes, a possible overflow of the allowable field lengths might result.

- You can specify functions such as MAX, MIN, COUNT, SUM, and AVG. An example of specifying AVG is shown below:

```
'EMPL_HOURLY_RATE < AVG(EMPL_HOURLY_RATE)'
```

- On an SQL CREATE user exec request, CA Telon uses the value in this field to specify a mass insert criterion, as in this example:

```
EXEC SQL INSERT INTO DSN8.TEMPL
      SELECT *                ** mass
      FROM DSN82.TEMPL        ** insert
      WHERE WORKDEPT='D11'    ** criterion
      END-EXEC.
```

IGNORE

Status codes to be ignored. All DBMS-specific status codes are valid. Generic status values are translated to the corresponding DBMS-specific status codes at generation time.

This table presents generic values and how CA Telon handles generic values.

Generic Status	What Is Ignored
OK	Blank or zero status codes
NOTFOUND or NFD	Not found status codes
DUPLICATE or DUP	Multiple record or key occurrences
LOGICERR or LOG	Errors on access that depends on prior required conditions
REF	Violations of referential integrity valid for DB2 only; SQLCODES -530 through -536
SECURITY or SEC	Security violations
ENDFILE or EOF	EOF conditions
NOTAVAIL or NAV	Conditions when resources are not available
DBMERROR or DBM	Any return code not specified above (SQL only)
ALL	Any return code

If CA Telon encounters a return code other than blanks that is not included in the IGNORE field for an I/O, CA Telon assumes ABEND processing as follows:

- For all I/O except BROWSE, CA Telon invokes ABEND processing. OK is the default.
- For BROWSE I/O, CA Telon does not invoke ABEND processing. ALL is the default.

IOAREA

The data area used for this I/O operation. The value in this field overrides the value in the COPYLV1 field.

See Update Database Segment for more information about the COPYLV1 field.

To request additional space for entry in this field, enter **U** in the select field in column 80. You are transferred to the Field Extension screen, where the field is redisplayed with additional lines for data entry.

If you want to select columns for the SENCOLS parameter, from the list of available columns, enter an **S** in the IOAREA flag field.

UPDATE

A value to specify whether an update is generated for the record that is current of the cursor. Values are:

Y

CA Telon generates an update for the current record of the cursor. CA Telon does not generate the statement to perform the update; you must do this in custom code. The name of the update paragraph follows standard user exec naming conventions, except that CA Telon appends -UPDATE or _UPDATE to the name.

N

(Default.) CA Telon does not generate an update for the current record of the cursor.

If both an UPDATE and a CPYCALL are coded for the same data access request, the UPDATE call is generated according to CA Telon defaults. Information coded in the CPYCALL is not used to generate the UPDATE call. If an UPDATE call must match information in a CPYCALL, define a separate UPDATE call with its own CPYCALL.

DELETE

A value to specify whether a delete is generated for the record that is current of cursor. Values are:

Y

CA Telon generates a delete for the current record of the cursor. CA Telon does not generate the statement to perform the delete; you must do this in custom code. The name of the delete paragraph follows standard user exec naming conventions, except that CA Telon appends -DELETE or _DELETE to the name.

N

(Default.) CA Telon does not generate a delete for the current record of the cursor.

HOLD CURSOR

Enter **Y** in this field to generate WITH HOLD in the DECLARE CURSOR statement for BROWSE, READNEXT, TRANSACT, MATCH, or MERGE data access request.

Values are:

Y

Generate WITH HOLD

N

(Default) Do not generate WITH HOLD.

JOINOPT

Determines the kind of join to be generated for this data access request.
Values are:

Spaces

(Default) Implicit inner join

L - Left

Left outer join

R - Right

Right outer join

F - Full

Full outer join

I - EXPLI

Explicit inner join

RESULTNR (SPBROWSE, SPTRNACT and SPRDNEXT Data Access Only)

The number of the result set produced by the called stored procedure. If only one result set is produced, the number is 1.

Note: If RESULTNR is specified, then RESULTPR must also be specified.

SENCOLS

The columns that you want to select to be returned to the program.

The value in this field overrides the value in the AC field for the columns on the Create/Update SQL Tables/TLNROWS screen (for the row to which the data access request corresponds).

You can specify any built-in functions such as MAX, COUNT, AVG, MIN, and SUM. Enclose the entire expression in single quotes. For example:
SENCOLS 'COUNT(*)'

If you need additional space for the SENCOLS field, enter **U** in the SENCOLS Flag field in column 80 to transfer to the Extension screen, where the field is redisplayed with additional lines for data entry.

If you want to select columns for the SENCOLS parameter, from the list of available columns, enter an **S** in the SENCOLS flag field.

To enter user-defined datatypes for the fields listed in the SENCOLS field, enter a **D** in the SENCOLS flag field.

ORDERBY

Use this field to list the columns to be included in the ORDERBY clause of the current data access. For a BROWSE request, the ORDERBY specification overrides the key settings specified for the table on the Create/Update SQL Tables/TLNROWS screen.

If you need additional space for the ORDERBY clause, enter **U** in the ORDERBY flag field in column 80 to transfer to the Field Extension screen.

If you want to select columns for the ORDERBY parameter, from the list of available columns, enter an **S** in the ORDERBY flag field.

GROUPBY

The columns to be included in the GROUPBY clause of the current data access request.

To request additional space for entry in this field, enter **U** in the select field in column 80. You are transferred to the Field Extension screen, where the field is redisplayed with additional lines for data entry.

If you want to select columns for the GROUPBY parameter, from the list of available columns, enter an **S** in the GROUPBY flag field.

HAVING

The columns to be included in the HAVING clause of the data access request.

If you want to select columns for the GROUPBY parameter, from the list of available columns, enter an **S** in the GROUPBY flag field.

KEYCOLS

The key columns in the I/O that CA Telon generates. The value in this field overrides the key name that can be set for the columns on the Create/Update SQL Tables/TLNROWS screen (for the row to which the I/O corresponds). Defaults, if used, are displayed followed by the at sign (@).

Note: The KEYCOLS field cannot be entered for BROWSE, READNEXT, and TRANSACT requests, and may be protected for a MATCH or MERGE request. For these data access requests, the ORDERBY field also functions as the KEYCOLS clause.

If you want to select columns for the KEYCOLS parameter, from the list of available columns, enter an **S** in the KEYCOLS flag field.

OPCODE

Use this field to define the relational operator for the data access request.

The value in this field overrides the OPCODE that CA Telon automatically generates (shown with the inheritance @ sign). Values are:

Value	Description
=	(Default) Equal to

¬=	Not equal to
>=	Greater than or equal to
<=	Less than or equal to
>	Greater than
<	Less than

Edit Flag fields

The custom code points on this screen (CPYCALL, CPYKEY, CPYINIT, CPYTERM, and RESLTCC for SPBROWSE, SPRDNEXT, and SPTRNACT) are each preceded by a one-position edit flag field. To create a new custom code member for one of these custom code points (or to have the TDF generate a data access request for CPYCALL), enter a **U** in the flag field of that custom code point. Once the custom code member named in the custom code member field has been created and saved, you can enter any non-blank character in the corresponding flag field to update the custom code. If you want to "open" the custom code point (that is, disassociate the named custom code member from the custom code point without deleting the custom code member itself), space out or erase the name in the custom code name field.

CPYCALL

The name of the custom code member into which the TDF should generate or has already generated the current data access request. You can modify the generated data access request as needed in this member. The code that the Generator would generate for this data access request is replaced by a COPY/%INCLUDE statement so that the CPYCALL custom code member is included in place of the Generator code.

To create the CPYCALL custom code member, enter a **U** in the CPYCALL flag field and name the custom code member. To update an existing CPYCALL custom code member, enter any non-blank character in the flag field.

To open the CPYCALL custom code point (disassociate the named custom code member from the custom code point without deleting the custom code member itself), space out or erase the name in the CPYCALL custom code name field.

Notes:

- CPYCALL should only be used with user exec data access requests such as, READ or CREATE. It is not recommended for use with auto exec calls such as BROWSE or OIREAD.
- When a CPYCALL copybook exists for this data access request, the CPYCALL custom code instead of the generated code, is displayed when PREVIEW is requested.

CPYINIT

The name of the custom code member to be copied into the program before code for the current data access request.

To create the CPYINIT custom code member, enter a **U** in the CPYINIT flag field and name the custom code member. You are presented with an empty custom code member. To update an existing CPYINIT custom code member, enter any non-blank character in the flag field.

To open the CPYINIT custom code point (disassociate the named custom code member from the custom code point without deleting the custom code member itself), space out or erase the name in the CPYINIT custom code name field.

Note: CPYINIT should only be used with user exec data access requests such as, READ or CREATE. It is not recommended for use with auto exec calls such as BROWSE or OIREAD.

CPYKEY

The name of the custom code member that overrides the WHERE clause generated by the Generator for the current data access request. You can modify the generated data access request as needed in this member. The code that the Generator would generate for this data access request is replaced by a COPY/%INCLUDE statement so that the PYCALL custom code member is included in place of the Generator code.

To create the CPYKEY custom code member, enter a **U** in the CPYKEY flag field and name the custom code member. You are presented with an empty custom code member. To update an existing CPYKEY custom code member, enter any non-blank character in the flag field.

To open the CPYKEY custom code point (disassociate the named custom code member from the custom code point without deleting the custom code member itself), space out or erase the name in the CPYKEY custom code name field.

Note: CPYKEY should only be used with user exec data access requests such as, READ or CREATE. It is not recommended for use with auto exec calls such as BROWSE or OIREAD.

RESULTCC (SPBROWSE, SPTRNACT and SPRDNEXT Data Access Only)

The name of the custom code member to be included at the beginning at the end of the S-200-CURSOR-*resltpr-nn*, paragraph for the current data access request, where *resltpr* is the name of the stored procedure being called and *nn*. is the result set number to be returned from the call to call to the stored procedure.

To create the RESULTCC custom code member, enter a **U** in the RESULTCC flag field and name the custom code member. You are presented with an empty custom code member. To update an existing RESULTCC custom code member, enter any non-blank character in the flag field.

To open the RESULTCC custom code point (disassociate the named custom code member from the custom code point without deleting the custom code member itself), space out or erase the name in the RESULTCC custom code name field.

Note: RESULTCC is only valid for calls to a stored procedure, such as SPBROWSE, SPTRNACT, and SPRDNEXT. The RESULTCC field and its flag field are only available on this screen for these three data access requests.

CPYTERM

The name of the custom code member to be copied into the program for the current data access request. :p To create the CPYTERM custom code member, enter a **U** in the CPYTERM flag field and name the custom code member. You are presented with an empty custom code member. To update an existing CPYTERM custom code member, enter any non-blank character in the flag field.

To open the CPYTERM custom code point (disassociate the named custom code member from the custom code point without deleting the custom code member itself), space out or erase the name in the CPYTERM custom code name field.

Note: CPYTERM should only be used with user exec data access requests such as, READ or CREATE. It is not recommended for use with auto exec calls such as BROWSE or OIREAD.

RESLTPR

The name of the stored procedure being called to return the result set referred to with the RESLTNR field.

Notes:

- RESLTPR is only valid for calls to a stored procedure such as SPBROWSE, SPTRNACT, and SPRDNEXT. The RESLTPR field and its flag field are only available on this screen for these three data access requests.
- If RESLTNR is specified, then RESLTPR must also be specified.
-

Get Diagnostics List

Access

On the Update DB2 Detail Data Access (S147) screen, entering a non-blank character in the GETDIAG flag field transfers you to the GET DIAGNOSTICS list screen.

Program ID

S240

Function

Allows you to define up to eight GET DIAGNOSTICS statements for any DB2 data access request.

The program to which the GET DIAGNOSTICS statements belong is displayed in the upper left corner of the screen.

TRCC2K.SD GET DIAGNOSTICS LIST *****

COMMAND ==> _____

DATA ACCESS REQUEST: AUTOEXEC BROWSE TRGEMPL

U	NO	TYPE	LOCFLAG	SVCODE	DIAGNOSTICS			
					GDCUST	GDOPNC	GDCLSC	CONDPTR
-	1	----	-	-	-----	-----	-----	-----
-	2	----	-	-	-----	-----	-----	-----
-	3	----	-	-	-----	-----	-----	-----
-	4	----	-	-	-----	-----	-----	-----
-	5	----	-	-	-----	-----	-----	-----
-	6	----	-	-	-----	-----	-----	-----
-	7	----	-	-	-----	-----	-----	-----
-	8	----	-	-	-----	-----	-----	-----

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

Data Access Request

This display-only field identifies the data access request to which this GET DIAGNOSTICS statement belongs.

U

Allows you to perform operations on the GET DIAGNOSTIC statement. Values are:

U

Update the GET DIAGNOSTICS statement. This option is not valid for CUST statements.

Z

ZAP (purge) the GET DIAGNOSTICS statement.

NO

The sequential number of the current GET DIAGNOSTICS statement. This is a display-only field. Values are 1 through 8.

TYPE

Identifies the type of GET DIAGNOSTIC statement this is. Values are:

STAT

Statement GET DIAGNOSTICE statement

CUST

Custom code only

COND

Condition or Connection GET DIAGNOSTICS statement

COMB

Combination GET DIAGNOSTICS statement

LOCFLAG

The location flag indicates where in the program the GET DIAGNOSTICS statement is generated. Values are:

I - IMBED

Generate the GET DIAGNOSTICS statement inline, immediately after the data access request.

G - G100

Generate the GET DIAGNOSTICS statement in a G-100 paragraph that can be performed from custom code.

C - COMMIT

Generate the GET DIAGNOSTICS statement immediately following the EXEC SQL COMMIT statement.

R - ROLLB

Generate the GET DIAGNOSTICS statement immediately following the EXEC SQL ROLLBACK statement

SVCODE

Indicates whether the SQLCODE from the data access request to which the current GET DIAGNOSTICS statement belongs is saved before execution of the current GET DIAGNOSTICS statement, then restored after it is performed. Enter a Y or N in this field.

DIAGNOSTICS

A comma-separated, paired list of diagnostic items being retrieved. The first member of the pair contains a 4-byte code indicating the information being retrieved, and the second member identifies the host variable name to which that information is assigned.

Note: You can choose to specify the mappings of diagnostic items to host variables using the S241, S242, and S243 detail screens. If this is done, and there are more than 54 bytes of information in the DIAGS specification, this field is protected, and any modifications will require use of the detail screen.

GDCUST

The name of custom code point immediately following the FETCH command for the BROWSE, TRANSACT, and READNEXT data access request to which the current GET DIAGNOSTICS statement belongs.

Note: For CUST GET DIAGNOSTICS, this is the only custom code point available.

GDOPNC

The name of custom code point immediately following the OPEN CURSOR command for the BROWSE, TRANSACT, and READNEXT data access request to which the current GET DIAGNOSTICS statement belongs.

Note: GDOPNC is not available for CUST GET DIAGNOSTICS statements.

GDCLSC

The name of custom code immediately following the CLOSE CURSOR command for the BROWSE, TRANSACT, and READNEXT data access request to which the current GET DIAGNOSTICS statement belongs.

Note: GDCLSC is not available for CUST GET DIAGNOSTICS statements.

CONDPTR

The number of the iteration of diagnostics the current GET DIAGNOSTICS statement returns. If a host variable is specified, you have the option of allowing CA Telon to generate a default host variable for CONDPTR by specifying a "@" sign. The CONDPTR value can be a hardcoded number or assigned to the host variable identified in this parameter.

Type	Default COBOL Name	Default PL/I Name
COND	GETDIAG-CONDITION-PTR	GETDIAG_CONDITION_PTR
COMB with Condition	GETDIAG-COMB-COND-PTR	GETDIAG_COMB_COND_PTR
COMB with Connection	GETDIAG-COMB-CONN-PTR	GETDIAG_COMB_CONN_PTR

The CONDPTR field applies only to COND and COMB types.

Note: The maximum number of iterations may be acquired from the NUMBER diagnostic of a Statement GET DIAGNOSTICS, which can be executed immediately before a Condition or Combination GET DIAGNOSTICS.

Get Diagnostics Statement Information

Access

To access the Statement GET DIAGNOSTICS screen, enter a **U** in front of a STMT GET DIAGNOSTICS entry on the GET DIAGNOSTICS List (S240) screen, then press Enter.

Program ID

S241

Function

Allows you to identify host variables for the parameters that you want to have returned by a Statement GET DIAGNOSTICS statement.

```
TRC256.BD  GET DIAGNOSTICS STATEMENT* *****
COMMAND ==> _____

GET DIAGNOSTICS # _ STATEMENT INFORMATION FOR _____

LOCFLAG _  SVCODE _  GDCUST _____

SPECIFY HOST VARIABLES FOR DESIRED ITEMS:

DIAG _____ DB2_GET_DIAGNOSTICS_DIAGNOSTICS
LSRW _____ DB2_LAST_ROW
MORE _____ MORE
NMPM _____ DB2_NUMBER_PARAMETER_MARKERS
NMRS _____ DB2_NUMBER_RESULT_SETS
NUMB _____ NUMBER
RCNT _____ ROW_COUNT
RTST _____ DB2_RETURN_STATUS
SACH _____ DB2_SQL_ATTR_CURSOR_HOLD
SACL _____ DB2_SQL_ATTR_CURSOR_SCROLLABLE
SACR _____ DB2_SQL_ATTR_CURSOR_ROWSET
SACS _____ DB2_SQL_ATTR_CURSOR_SENSITIVITY
SACT _____ DB2_SQL_ATTR_CURSOR_TYPE
```

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

NO

The sequential number of the current GET DIAGNOSTICS statement. This is a display-only field. Values are 1 through 8.

STATEMENT INFORMATION FOR

This display-only field identifies the data access request to which this GET DIAGNOSTICS statement belongs.

LOCFLAG

The location flag indicates where in the program the GET DIAGNOSTICS statement is generated. Values are:

I - IMBED

Generate the GET DIAGNOSTICS statement inline, immediately after the data access request.

G - G100

Generate the GET DIAGNOSTICS statement in a G-100 paragraph that can be performed from custom code.

C - COMMIT

Generate the GET DIAGNOSTICS statement immediately following the EXEC SQL COMMIT statement.

R - ROLLB

Generate the GET DIAGNOSTICS statement immediately following the EXEC SQL ROLLBACK statement.

SVCODE

Indicates whether the SQLCODE from the data access request to which the current GET DIAGNOSTICS statement belongs is saved before execution of the current GET DIAGNOSTICS statement, then restored after it is performed. Enter a **Y** or **N** in this field.

GDCUST

The name of custom code point immediately following the FETCH command for the BROWSE, TRANSACT, and READNEXT data access request to which the current GET DIAGNOSTICS statement belongs.

Note: For CUST GET DIAGNOSTICS, this is the only custom code point available.

GDOPNC

The name of custom code point immediately following the OPEN CURSOR command for the BROWSE, TRANSACT, and READNEXT data access request to which the current GET DIAGNOSTICS statement belongs.

Note: GDOPNC is not available for CUST GET DIAGNOSTICS statements.

GDCLSC

The name of custom code immediately following the CLOSE CURSOR command for the BROWSE, TRANSACT, and READNEXT data access request to which the current GET DIAGNOSTICS statement belongs.

Note: GDCLSC is not available for CUST GET DIAGNOSTICS statements.

HOST VARIABLE

Host variable into which the values returned for the specific diagnostic from the current GET DIAGNOSTICS statement are stored. To indicate that you want the default host variable for a diagnostic, enter an @ in this field.

Note: At least one host variable must be specified for a GET DIAGNOSTICS statement.

Get Diagnostics Condition Information

Access

To access the first COND GET DIAGNOSTICS screen, enter a **U** in front of a COND GET DIAGNOSTICS entry on the GET DIAGNOSTICS List (S240) screen, then press Enter. You can use PF7 (page backward) and PF8 (page forward) to toggle between the three pages of this screen.

Program ID

S242 (3 pages)

Function

Allows you to identify host variables for the parameters you want to have returned by a Statement GET DIAGNOSTICS statement.

```
HHIIII.DD  GET DIAGNOSTICS STATEMENT * *****
COMMAND ==> _____ PAGE 01

GET DIAGNOSTICS # _  CONDITION INFORMATION FOR _____

LOCFLAG _  SVCODE _  GDCUST _____  GDOPNC _____  GDCLSC _____
CONDPTR _____

SPECIFY HOST VARIABLES FOR DESIRED ITEMS:

      AUTD _____  DB2_AUTHORIZATION_ID
      AUTY _____  DB2_AUTHENTICATION_TYPE
      CNSS _____  DB2_CONNECTION_STATUS
      CNST _____  DB2_CONNECTION_STATE
      ENTY _____  DB2_ENCRYPTION_TYPE
      PROI _____  DB2_PRODUCT_ID
      SVCN _____  DB2_SERVER_CLASS_NAME
      CANM _____  CATALOG_NAME
      CONM _____  CONDITION NUMBER
      CRNM _____  CURSOR NAME
      ERR1 _____  DB2_ERROR_CODE1
      ERR2 _____  DB2_ERROR_CODE2
      ERR3 _____  DB2_ERROR_CODE3
      ERR4 _____  DB2_ERROR_CODE4
```

```
TRC256.BD  GET DIAGNOSTICS STATEMENT * *****
COMMAND ==> _____ PAGE 02
```

```
GET DIAGNOSTICS # _  CONDITION INFORMATION FOR _____
```

```
LOCFLAG _  SVCODE _  GDCUST _____  GDOPNC _____  GDCLSC _____
CONDPTR _____
```

```
SPECIFY HOST VARIABLES FOR DESIRED ITEMS:
```

IEPT _____	DB2_INTERNAL_ERROR_POINTER
MODE _____	DB2_MODULE_DETECTING_ERROR
MSGI _____	DB2_MESSAGE_ID
MTXT _____	MESSAGE_TEXT
RCDE _____	DB2_REASON_CODE
ROWN _____	DB2_ROW_NUMBER
RSQC _____	DB2_RETURNED_SQLCODE
RSST _____	RETURNED_SQLSTATE
SQL1 _____	DB2_SQLERRD1
SQL2 _____	DB2_SQLERRD2
SQL3 _____	DB2_SQLERRD3
SQL4 _____	DB2_SQLERRD4
SQL5 _____	DB2_SQLERRD5
SQL6 _____	DB2_SQLERRD6

```
HHI111.XX  GET DIAGNOSTICS STATEMENT * *****
COMMAND ==> _____ PAGE 03
```

```
GET DIAGNOSTICS # _  CONDITION INFORMATION FOR _____
```

```
LOCFLAG _  SVCODE _  GDCUST _____  GDOPNC _____  GDCLSC _____
CONDPTR _____
```

```
SPECIFY HOST VARIABLES FOR DESIRED ITEMS:
```

SSET _____	DB2_SQLERRD_SET
SVNM _____	SERVER_NAME
TKCT _____	DB2_TOKEN_COUNT
OT01 _____	DB2_ORDINAL_TOKEN_01
OT02 _____	DB2_ORDINAL_TOKEN_02
OT03 _____	DB2_ORDINAL_TOKEN_03
OT04 _____	DB2_ORDINAL_TOKEN_04
OT05 _____	DB2_ORDINAL_TOKEN_05
OT06 _____	DB2_ORDINAL_TOKEN_06
OT07 _____	DB2_ORDINAL_TOKEN_07
OT08 _____	DB2_ORDINAL_TOKEN_08
OT09 _____	DB2_ORDINAL_TOKEN_09
OT10 _____	DB2_ORDINAL_TOKEN_10
OT11 _____	DB2_ORDINAL_TOKEN_11

Field Definitions

COMMAND

For information, see Primary Commands.

NO

The sequential number of the current GET DIAGNOSTICS statement. This is a display-only field. Values are 1 through 8.

CONDITION INFORMATION FOR

This display-only field identifies the data access request to which this GET DIAGNOSTICS statement belongs.

LOCFLAG

The location flag indicates where in the program the GET DIAGNOSTICS statement is generated. Values are:

I - IMBED

Generate the GET DIAGNOSTICS statement inline, immediately after the data access request.

G - G100

Generate the GET DIAGNOSTICS statement in a G-100 paragraph that can be performed from custom code.

C - COMMIT

Generate the GET DIAGNOSTICS statement immediately following the EXEC SQL COMMIT statement.

R - ROLLB

Generate the GET DIAGNOSTICS statement immediately following the EXEC SQL ROLLBACK statement.

SVCODE

Indicates whether the SQLCODE from the data access request to which the current GET DIAGNOSTICS statement belongs is saved before execution of the current GET DIAGNOSTICS statement, then restored after it is performed. Enter a **Y** or **N** in this field.

GDCUST

The name of custom code point immediately following the FETCH command for the BROWSE, TRANSACT, and READNEXT data access request to which the current GET DIAGNOSTICS statement belongs.

Note: For CUST GET DIAGNOSTICS, this is the only custom code point available.

GDOPNC

The name of custom code point immediately following the OPEN CURSOR command for the BROWSE, TRANSACT, and READNEXT data access request to which the current GET DIAGNOSTICS statement belongs.

Note: GDOPNC is not available for CUST GET DIAGNOSTICS statements.

GDCLSC

The name of custom code immediately following the CLOSE CURSOR command for the BROWSE, TRANSACT, and READNEXT data access request to which the current GET DIAGNOSTICS statement belongs.

Note: GDCLSC is not available for CUST GET DIAGNOSTICS statements.

CONDPTR

The number of the iteration of diagnostics the current GET DIAGNOSTICS statement returns. You have the option of allowing CA Telon to generate a default host variable for CONDPTR by specifying an "@" sign. The CONDPTR value can be a hardcoded number or be assigned to the host variable identified in this parameter.

Type	Default COBOL Name	Default PL/I Name
COND	GETDIAG-CONDITION-PTR	GETDIAG_CONDITION_PTR

HOST VARIABLE

Host variable into which the values returned for the specific diagnostic from the current GET DIAGNOSTICS statement are stored. To indicate that you want the default host variable for a diagnostic, enter an @ in this field. At least one host variable must be specified for a GET DIAGNOSTICS statement.

Note: The first seven items listed on Page 1 of this screen are Connection diagnostic items. The remainder on this and subsequent pages are Condition items.

Also, the maximum number of iterations may be acquired from the NUMBER diagnostic of a Statement GET DIAGNOSTICS that can be executed immediately before a Condition GET DIAGNOSTICS.

Get Diagnostics Combined Information

Access

To access the Combined GET DIAGNOSTICS screen, enter a **U** in front of a COMB GET DIAGNOSTICS entry on the GET DIAGNOSTICS List (S240) screen, then press Enter.

Program ID

S243

Function

Collects information for a Combined GET DIAGNOSTICS statement. A Combined statement returns all values from selected statement types.

```
TRC256.BD  GET DIAGNOSTICS STATEMENT* *****  
COMMAND ==> _____  
  
GET DIAGNOSTICS # _  COMBINED INFORMATION FOR _____  
  
LOCFLAG _  SVCODE _  GDCUST _____  GDOPNC _____  GDCLSC _____  
  
RECEIVING HOST VARIABLE  
COMBINATION TYPE:  _ STMT  _ COND  _ CONN  
  
FOR CONDITION AND CONNECTION:  
  
CONDPTR _____
```

Field Definitions

COMMAND

For information, see Primary Commands.

NO

This display-only field is the sequential number of the current GET DIAGNOSTICS statement. Values are 1 through 8.

COMBINED INFORMATION FOR

This display-only field identifies the data access request to which this GET DIAGNOSTICS statement belongs.

LOCFLAG

The location flag indicates where in the program the GET DIAGNOSTICS statement is generated. Values are:

I - IMBED

Generate the GET DIAGNOSTICS statement inline, immediately after the data access request.

G - G100

Generate the GET DIAGNOSTICS statement in a G-100 paragraph that can be performed from custom code.

C - COMMIT

Generate the GET DIAGNOSTICS statement immediately following the EXEC SQL COMMIT statement.

R - ROLLB

Generate the GET DIAGNOSTICS statement immediately following the EXEC SQL ROLLBACK statement

SVCODE

Indicates whether the SQLCODE from the data access request to which the current GET DIAGNOSTICS statement belongs is saved before execution of the current GET DIAGNOSTICS statement, then restored after it is performed. Enter a **Y** or **N** in this field.

GDCUST

The name of custom code point immediately following the FETCH command for the BROWSE, TRANSACT, and READNEXT data access request to which the current GET DIAGNOSTICS statement belongs.

Note: For CUST GET DIAGNOSTICS, this is the only custom code point available.

GDOPNC

The name of custom code point immediately following the OPEN CURSOR command for the BROWSE, TRANSACT, and READNEXT data access request to which the current GET DIAGNOSTICS statement belongs.

Note: GDOPNC is not available for CUST GET DIAGNOSTICS statements.

GDCLSC

The name of custom code immediately following the CLOSE CURSOR command for the BROWSE, TRANSACT, and READNEXT data access request to which the current GET DIAGNOSTICS statement belongs.

Note: GDCLSC is not available for CUST GET DIAGNOSTICS statements.

RECEIVING HOST VARIABLE

VARCHAR host variable in which a string of all values returned from the Combined GET DIAGNOSTICS statement are stored. You have the option of allowing CA Telon to generate a default receiving host variable by specifying an "@" sign.

Type	Default COBOL Name	Default PL/I Name
COMB with Condition	GETDIAG-COMB-COND-HOSTV	GETDIAG_COMB_COND_HOSTV
COMB with Connection	GETDIAG-COMB-CONN-HOSTV	GETDIAG_COMB_CONN_HOSTV

COMBINATION TYPE

The types of GET DIAGNOSTICS you want included in this Combined GET DIAGNOSTICS statement. You can select one of three types.

STMT

Load Statement diagnostics

COND

Load Condition diagnostics

COMB

Load Connection diagnostics

CONDPTR

The number of the iteration of diagnostics the current GET DIAGNOSTICS statement returns. You have the option of allowing CA Telon to generate a default receiving host variable by specifying an "@" sign. The CONDPTR value can be a hardcoded number or be assigned to the host variable identified in this parameter.

Type	Default COBOL Name	Default PL/I Name
COMB with Condition	GETDIAG-COMB-COND-PTR	GETDIAG_COMB_COND_PTR
COMB with Connection	GETDIAG-COMB-CONN-PTR	GETDIAG_COMB_CONN_PTR

Note: The maximum number of iterations may be acquired from the NUMBER diagnostic of a Statement GET DIAGNOSTICS that can be executed immediately before a Combination GET DIAGNOSTICS.

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

FETCH DETAILS FOR

This display-only field identifies the data access request.

SENSITIVITY

Sensitivity type for the FETCH for the current data access request. Only one value can be selected. Values are:

- SENSITIVE
- INSENSITIVE
- ASENSITIVE
- SENSSDYN
- SENSSTAT
- NOScroll

Notes:

- The sensitivity parameter allows you to select a scrollable cursor option for the current data access request. If you do not select an option or you select the NOScroll option, the cursor is not scrollable. To make your cursor scrollable, select one of the other sensitivity options. Each of the sensitivity options (including no specified option) has an impact on the generated DECLARE CURSOR and FETCH statements for the current data access request.
- If you have two data access requests for the same table, and both have a sensitivity option selected, only the option from the first data access request is used to generate the EXEC SQL DECLARE...CURSOR statement unless other data access requests have a label.

FETCH ORIENTATION

You can choose from a variety of FETCH orientation options. You can only select one option. You must also decide whether multiple rows will be retrieved by the current data access request.

For a single-row FETCH, the following values refer to a single-row. For a multi-row FETCH, they refer to the rowset to be returned.

BEFORE

Applies only to single-row FETCH

AFTER

Applies only to single-row FETCH

NEXT**PRIOR****FIRST****LAST****CURRENT****ABSOLUTE**

If you select this value, you must specify either a number or a host variable to contain the number in the field following ABSOLUTE. This number or host variable identifies the row to be fetched.

You have the option of allowing CA Telon to generate a default host variable for ABSOLUTE by specifying an "@" sign.

Default COBOL Name	Default PL/I Name
FTCHOPT-ROW-NUMBER	FTCHOPT_ROW_NUMBER

RELATIVE

If you select this value, you must specify either a number or a host variable to contain the number in the field following RELATIVE. This number or host variable identifies the row to be fetched.

You have the option of allowing CA Telon to generate a default host variable for RELATIVE by specifying an "@" sign.

Default COBOL Name	Default PL/I Name
FTCHOPT-ROW-NUMBER	FTCHOPT_ROW_NUMBER

FOR...ROWS

You must specify the number of rows to fetch for multi-row fetches. The value in this parameter can be a hard-coded number or a host variable that contains the number of rows to be fetched.

You have the option of allowing CA Telon to generate a default host variable for FOR...ROWS by specifying an "@" sign.

Default COBOL Name	Default PL/I Name
FTCHOPT-MULTROW-COUNT	FTCHOPT_MULTROW_COUNT

OPTIMIZE FOR nn ROWS

The number of rows specified in an OPTIMIZE parameter for generated SQL data access requests with a cursor (BROWSE, READNEXT, TRANSACT, MATCH, and MERGE).

FETCH ONLY

Specifies whether FOR FETCH ONLY should be generated in the DECLARE CURSOR statement for generated DB2 calls with a cursor (BROWSE, READNEXT, TRANSACT, MATCH, and MERGE). Values are:

N

(Default.) Do not generate FOR FETCH ONLY

Y

Generate FOR FETCH ONLY

ISOLATION

Use this parameter to identify the isolation level at which the data access request is executed. Values are:

TDF	Generated	Meaning
CS	CS	Cursor stability
UR	UR	Uncommitted read
RR	RR	Repeatable read
RRK	RR KEEP UPDATE LOCKS	Repeatable read keep update locks
RS	RS	Read stability
RSK	RS KEEP UPDATE LOCKS	Read stability keep update locks

ALTCURS

Use to specify an alternate cursor if you want to use one for the current data access request. An alternate cursor must be defined for another data access request in the current data access request.

Notes:

- The TDF does not validate the value of this parameter to ensure that the cursor exists.
- The CURSOR name is made up of two parts connected with a hyphen: the data access request label or type (for example, READNEXT) and the name of the table being accessed.

Field Extension

Access

On the Update SQL Detail Data Access screen, enter **U** in the select field in column 80 for one of these fields:

- KEY OR WHERE
- IOAREA
- SENCOLS
- ORDERBY
- GROUPBY
- KEYCOLS

Program ID

S144

Function

Add data to certain fields that originate on the Update SQL Detail Data Access screen.

Note: Any values valid for the field type (for example, IOAREA, GROUPBY) that corresponds to the field on this screen are also valid on this screen.

See the *Programming Concepts Guide* for information on data inheritance.

SQL FIELD EXTENSION SCREEN		*****	
COMMAND ==> _____			
	TRC210.BD	AUTOEXEC	TRANSACTION TRGEMPL ***
	TABLE NAME TELON.TRGEMPL		ROW NAME TRGEMPL
SENCOLS	_____		A

UPDCOLS	_____		A

Field Definitions

COMMAND

For information, see Primary Commands.

Field Type

A definition for the current field type (KEY or WHERE, IOAREA, SENCOLS, ORDERBY, GROUPBY, KEYCOLS).

(Selection field) A

A select field in column 80. To select the names of columns to be added to the field rather than keying them in, enter an **S** to the Select Columns (S187) screen, which lists valid columns you can select for this transaction. After selecting columns that you want to add, press PF3 to return to the screen.

For SENCOLS only, you can also transfer to the Update User-Defined Datatype (S184) screen by entering a **D** in the selection field.

UPDCOLS

A list of columns you want updated with this data access.

This field is displayed only when you request an update from the Select Columns screen by entering **U** in the SENCOLS field and **Y** in the UPDATE field.

Update User-Defined Datatypes

Access

On either the Update SQL Detail Data Access (S147) or the Field Extension (S144) screen enter a **D** in the SENCOLS flag field.

Program ID

S184

TYPE

(Display-only.) The SENCOLS field type. Values for TYPE, LTH, and DEC are listed in the following table:

SQL Data Type	CA Telon Field			Description of Data Type
	TYPE	LTH	DEC	
CHAR (<i>n</i>)	CHAR	<i>n</i>		A fixed-length character string of length <i>n</i> where <i>n</i> is in the range 1-254.
DASH				A TLNROW.
DATE	DATE			A date value in either of two forms: U.S. or International.
DECIMAL (<i>p</i> , <i>s</i>)	DEC	<i>p</i>	<i>s</i>	A decimal number. The first integer, <i>p</i> , is the precision of the number (the total number of digits) and must be in the range 1 to 15. The second integer, <i>s</i> , is the scale of the number (the number of digits to the right of the decimal point) and must be in the range 0 to <i>p</i> .
DOUBLE PRECISION	DBLPREC			A 64-bit (long) floating-point value with a seven-bit exponent and a binary precision of 56. The length of a DOUBLE PRECISION value is eight bytes.
FLOAT (<i>p</i>)	FLOAT	<i>p</i>		A floating-point value with a 7-bit exponent and a user-specified precision (<i>p</i>) to indicate the binary precision of the FLOAT value. Precision is an integer in the range 1 through 56. If the precision is greater than 24, the length of a FLOAT value is eight bytes otherwise the length is four bytes.
GRAPHIC (<i>n</i>)	GR	<i>n</i>		A fixed-length string of double-byte characters of length <i>n</i> where <i>n</i> is in the range 1-127.
INTEGER	INT			A long integer.

LONG INTEGER	LONGINT			A 64-bit signed long integer value. The length of a LONG INTEGER value is eight bytes (CA-IDMS/SQL only).
NUMERIC (p, s)	NUMERIC	p	s	A fixed-point, signed zoned decimal value. The first integer, p, is the precision of the number (the total number of digits) in the range 1 to 32. The second integer, s, is the scale of the number of digits to the right of the decimal point) in the range 0 to p.
REAL	REAL			A 32-bit (short) floating-point value with a seven-bit exponent and a binary precision of 24. The length of a REAL value is four bytes.
SMALLINT	SINT			A small integer.
TIME	TIME			A time value.
TIMESTAMP	STMP			The timestamp.
UNASSIGNED DECIMAL(p,s)	UNSGNDEC	p	s	A fixed-point, unsigned packed decimal value. The first integer, p, is the precision of the number (the total number of digits) in the range 1 to 31. The second integer, s, is the scale of the number of digits to the right of the decimal point) in the range 0 to p (CA-IDMS/SQL only).
UNSIGNED NUMERIC (p,s)	UNSGNNUM	p	s	A fixed-point, unsigned packed decimal value. The first integer, p, is the precision of the number (the total number of digits) in the range 1 to 32. The second integer, s, is the scale of the number of digits to the right of the decimal point) in the range 0 to p (CA-IDMS/SQL only).
VARCHAR (n)	VCHR	n		A varying-length character string of maximum length n that must be in the range 1 to 32,767.
VARGRAPHIC (n)	VGR	n		A varying-length string of double-byte characters of maximum length n that must be in the range 1 to 32,767.

LTH

The length, if the type is variable length, or the precision, if the type is DECIMAL. See the table in the TYPE field description in this section.

DEC

The scale, if the type is DECIMAL. See the table in the TYPE field description in this section.

Select Columns

Access

- On the Update SQL Detail Data Access screen, enter **S** in the selection field in column 80.
- On the SQL Field Extension screen, enter any character in the selection field in column 80.

Program ID

S187

Function

Lists the columns in the table.

SELECT COLUMNS *****									
COMMAND ==>								PAGE	__
TABLE NAME		ROW NAME							
S	COLUMN NAME	ALIAS	KY/C	TYPE	LTH/DEC	N			
*	*****	*****	**	*	****	****	**	*	

Field Definitions

COMMAND

For information, see Primary Commands.

TABLE NAME

(*Protected field.*) Identifies the qualifier and name of the table. This value is from either the SQL catalog or the Create/Update SQL Tables/TLNROWS screen.

ROW NAME

(*Protected field.*) Identifies a row of the table.

DICTNAM (CA-IDMS/SQL Only)

(*Protected field.*) Displays the CA-IDMS Dictionary name from which a CA-IDMS/SQL Table definition was extracted. It is only displayed for CA-IDMS/SQL tables.

SCHEMA (CA-IDMS/SQL and CA-Datcom/SQL Only)

(*Protected field.*) Displays the schema name of the Table. It is only displayed for CA-IDMS or CA-Datcom/SQL tables.

S

A select field that enables you to place the corresponding column name in the list as designated by the Update SQL Detail Data Access screen. Values include any nonblank character.

Pressing End or Enter causes the TDF to append the selections to the list. You can select a column more than once by pressing Enter and reselecting the desired column.

COLUMN NAME

(*Protected field.*) Identifies the columns associated with the rows. The value corresponds to one in the COLUMN NAME field on the Create/Update SQL Tables/TLNROWS screen. The 18 byte maximum is displayed for all tables except CA-IDMS and CA-Datcom tables. For CA-IDMS/SQL and CA-Datcom/SQL tables only the first 25 bytes are displayed.

8. ALIAS

(*Protected field.*) Identifies one of the following:

- An I/O area for the column, which CA Telon uses in place of the COLUMN NAME value for the I/O that it generates
- The name of the TLNROW that you are defining, which CA Telon uses in the user exec procedure or paragraph names that it generates

The value was specified in the ALIAS field on the Create/Update SQL Tables/TLNROWS screen.

By entering DATATYPE on the command line, the ALIAS column changes into a DATATYPE column where DATATYPES for which the columns are to be CAST into can be entered. By entering ALIAS on the command line, the screen is redisplayed with the ALIAS information. In this manner, you can toggle between the two versions of the screen display.

KY (KEY)

(*Protected field.*) Specifies whether the column is included in generated I/O WHERE conditions. Possible values are:

Y

The column is a key column

P

The column is a key column in position *p* in the key column list

(Blank)

The column is not a key column

The value was specified in the KY field on the Create/Update SQL Tables/TLNROWS screen.

AC (ACCESS)

(*Protected field.*) Specifies whether the column is accessed by the EXEC SQL statement.

Possible values are:

Y

The column is accessed by the EXEC SQL statement

N

The column is not accessed by the EXEC SQL statement

The value was specified in the AC field on the Create/Update SQL Tables/TLNROWS screen.

TYPE

(*Protected field.*) Displays the data type of the column. The value was specified in the TYPE field on the Create/Update SQL Tables/TLNROWS screen. See Create/Update SQL Tables/TLNROWS/Temporary Tables for a full description of these options.

LTH

(*Protected field.*) Displays the length of a variable-length data type or the precision of a decimal data type. The value was specified in the LTH field on the Create/Update SQL Tables/TLNROWS screen.

13. DEC

(*Protected field.*) Displays the scale of a decimal data type. The value was specified in the DEC field on the Create/Update SQL Tables/TLNROWS screen.

^N (^NULL)

(*Protected field.*) Indicates whether the column is defined as not null. The value was specified in the –NU field on the Create/Update SQL Tables/TLNROWS screen. Possible values are:

Y

The column is defined as not null (null values are not allowed)

N

The column is defined such that null values are allowed

Update DL/I Detail Data Access

Access

On the Create/Update Data Group screen, enter **U** as a line command for a DL/I data access value.

Program ID

S145

Function

Updates I/O for the segment identified on the Create/Update Data Group screen by defining I/O characteristics not available on the previous screen:

- Characteristics valid for any type of I/O
- Characteristics valid for DL/I only

XXXXXX.XX ** X-XXX-XXXXXXXX-XXXXXX *****

COMMAND ==>

OPTIONS ==> PREVIEW __

GENERAL: FUNC _____ SSALIST _ (Y/N) CONCATK _ (Y/N)

* IGNORE _____

* IOAREA _____

* PATH _____ PARENTG _____ LOCKED _____ CURRENT _____

** DSCREF SEGMENT CMND IMSKEY OP KEY

A _____

Field Definitions

COMMAND

For information, see Primary Commands.

You can also enter the INIT command to reinitialize all segment characteristics for all database segments, including those not displayed on the screen.

PREVIEW

A facility that permits you to see the effects of varying fields upon the call without having to wait until CA Telon generates the program. Invoke PREVIEW by specifying a non-blank character.

FUNC

The four-byte DL/I function code that specifies the nature of the call, in conjunction with the value specified for the REQUEST field on the Create/Update Data Group screen. For example:

LABEL Field	REQUEST Field	FUNC Field
READ	READ	GU
READGHU	READ	GHU
READGHN	READ	GHN
READNEXT	READ	GU
UPDATE	UPDATE	(Defaults to GHU, REPL)
REPLACE	UPDATE	REPL
DELETE	DELETE	(Defaults to let GHU DLET)
DELETE2	DELETE	DLET

Note: For EXEC DLI, GET HOLD function codes are invalid.

SSALIST

A field that allows you to access the SSALIST field entries on the Update Database User I/O SSALIST screen. Values are:

Y

Access the screen

N

Do not access the screen

CONCATK

A field that allows you to specify whether to use qualification statements or a concatenated key to qualify an I/O request. This is equivalent to the **C** command code. Values are:

Y

Generate the **C** command code

N

Do not generate the **C** command code

IGNORE

Status codes to be ignored. All DBMS-specific status codes are valid. Generic status values are translated to the corresponding DBMS-specific status codes at generation time.

This table presents generic values and how CA Telon handles generic values:

Generic Status	What Is Ignored
OK	Blank or zero status codes
NOTFOUND or NFD	Not found status codes
DUPLICATE or DUP	Multiple record or key occurrences
LOGICERR or LOG	Errors on access that depends on prior required conditions
SECURITY or SEC	Security violations
ENDFILE or EOF	EOF conditions
NOTAVAIL or NAV	Conditions when resources are not available
ALL	Any return code

If CA Telon encounters a return code other than blanks that is not included in the IGNORE field for an I/O, CA Telon assumes ABEND processing as follows:

- For all I/O except BROWSE, CA Telon invokes ABEND processing. OK is the default.
- For BROWSE I/O, CA Telon does not invoke ABEND processing. ALL is the default.

IOAREA

The data area used for this I/O operation. The value in this field overrides the value in the COPYLBL field.

See Update Database Segment for more information about the COPYLBL field.

PATH

As part of the path call, the segment at (and below) the one CA Telon is to retrieve.

Entering a value in this field is equivalent to the **D** command code (for READ type data access) or an **N** command code (for REPL type data access) for all segment levels at (and below) the specified segment.

PARENTG

The segment that is the current parent level for succeeding calls.

The value in this field is equivalent to the EXEC DLI SETPARENT option and the P SSA command code.

LOCKED

A lock on the specified segment reserving it for exclusive use by the program, regardless of whether the program updates it. DL/I prevents the segment from being altered by other programs until the next program reaches a sync point.

The value in this field is equivalent to the Q SSA command code.

CURRENT

The segment above which CA Telon is to maintain position for all levels.

The value in this field is equivalent to the **V** SSA command code.

(Line command) A

A line command here allows you to request a list of SSAs or commands for the segment. Enter **U** or **S**.

The Update DLIDSCs For Segment Member screen is displayed, providing a list of all user-defined SSAs or DLIDSCs. Enter **S** in the COMMAND field on this screen to select that DLIDSC. CA Telon updates all available information on the Select New Row Name screen or the Update DL/I Detail Data Access screen, setting the segment or user I/O SSA values. The USECNT value is incremented at this time.

DSCREF

The SSA or qualification statements for CA Telon to use to build SSAs or qualification statements for this request. Values are:

****DFLT****

CA Telon uses the default SSA/command for this level. The default SSA is generated as a qualified SSA in most cases. However, the default SSA is generated as an unqualified SSA when the segment is unkeyed or the data group contains an I/O request for this segment with the DSCREF=**QUAL**. Therefore, if you use **QUAL** as a DSCREF in any I/O anywhere within that segment, the default DSCREF is unqualified.

QUAL

CA Telon uses the qualified SSA for this level and generates the same SSA as the **DFLT** qualified SSA.

UNQUAL

CA Telon uses the unqualified SSA for this level.

To increment USECNT appropriately, use the **U** or **S** command code to request a list of SSAs (the List SSAs screen) and then use the **S** command code to select DLIDSC for this segment level or I/O request.

SEGMENT

(Protected field.) Identifies the segment that you are listing.

CMND

The four-byte SSA command code that CA Telon defines on the segment it generates.

CA Telon prefixes the character or characters you enter with an asterisk (*) and adds a suffix of one to three hyphens (-) to make the four-byte code. For example, if you enter **D**, CA Telon would generate the SSA command code field as:

*D- -

If you do not specify a code, CA Telon uses the IMS default *---.

For EXEC DLI, you can use this field to specify segment qualification options. Note that EXEC DLI does not support all command codes.

IMSKEY

(Protected field.) Identifies the name of the key for the segment. For the **DFLT** SSA, the field is blank for a non-keyed segment. The value in this field defines the DBD FIELD statement, NAME=. If a custom SSA is chosen, then whatever SRCHFLD has been specified displays as the IMSKEY protected value. For a boolean custom SSA, it is the SRCHFLD of the first qualification that displays.

OP

For DL/I processing, the relational operator for the SSA that CA Telon generates.

If not specified, the default is ">=" for BROWSE data access and "=" for keyed access. For DL/I processing, values are:

Value	Description
=, EQ	Equal to
>=, GE	Greater than or equal to
=>, GE	Greater than or equal to
<=, LE	Less than or equal to
=<, LE	Less than or equal to
>, GT	Greater than
<, LT	Less than
¬=, NE	Not equal to
=¬, NE	Not equal to

Note: OP does not apply for unqualified SSAs; that is, for BROWSE requests for which there is no starting key value (STBRKEY, defined on the Create/Update File Segloop screen). If you use the SCHFLDx field on the Create or Update File SEGLOOP screen, OP can be (but need not always be) set to the equal sign (=).

These rules apply to OPCODE overrides:

- User I/O OPCODE overrides the segment level OPCODE
- Segment-level OPCODE overrides the DLIDSC OPCODE in Option 2

CA Telon generates the value of the *segment-name*-SSA-OPCODE in working storage based on the segment-level OPCODE override.

If you override the OPCODE at the user I/O, before the DL/I call in U-100, CA Telon moves the overridden OPCODE to xxxx-SSA-OPCODE and resets it back to the OPCODE in working storage after the DL/I call.

If the user I/O OPCODE is the same as the segment-level OPCODE, CA Telon does not generate MOVE statements for the OPCODE.

KEY

The host variable name whose value is moved in the SSA for the database call.

Update Database User I/O SSALIST

Access

On the Update DL/I Detail Data Access screen, enter **Y** in the SSALIST field.

Program ID

S185

Function

Specifies the list of host variables used as SSAs in the call to CBLTDLI, CEETDLI, or PLITDLI.

```

XXXXXX.XX  ** X-XXX-XXXXXXXX-XXXXXX  ***** SSALIST PROCESSING FORMAT ACTIVE
COMMAND ==>
OPTIONS ==> PREVIEW _

GENERAL: FUNC      _____  SSALIST _ (Y/N)
*          IGNORE  _____
*          IOAREA  _____

** SEGMENT  SSA LIST HOST VARIABLE
_____
  
```

Field Definitions

COMMAND

For information, see Primary Commands.

PREVIEW

A value to request a view of the generated call. Values include any nonblank character.

FUNC

The nature of the call, in conjunction with the value specified in the LABEL and REQUEST fields on the Create/Update Data Group screen. For example:

LABEL Field	REQUEST Field	FUNC Field
READ	READ	GU
READGHU	READ	GHU
READGHN	READ	GHN
READNEXT	READ	GU

UPDATE	UPDATE	(Defaults to GHU, REPL)
REPLACE	UPDATE	REPL
DELETE	DELETE	(Defaults to GHU, DLET)

For DL/I, the value in this field specifies a four-byte function code (GN, GNP, and so on). For EXEC DLI, *Get Hold* function codes are invalid.

SSALIST

A value to request transfer to another screen. Values are:

Y

Transfer to the Update DL/I Detail Data Access screen

N

Do not transfer to the Update DL/I Detail Data Access screen

(blank)

Do not transfer to the Update DL/I Detail Data Access screen

IGNORE

Status codes to be ignored. All DBMS-specific ignore values are valid. These generic ignore values are translated to the corresponding DBMS-specific value at generation time:

Status Code	What CA Telon Ignores
OK	Blank or zero status codes
NOTFOUND or NFD	Not found status codes
DUPLICATE or DUP	Multiple record/key occurrences
LOGICERR or LOG	Errors on accesses that are dependent on prior required conditions
SECURITY or SEC	Security violations
ENDFILE or EOF	EOF conditions
NOTAVAIL or NAV	Conditions when resources are not available
ALL	All conditions including OK

If CA Telon encounters a return code (other than blanks) that is not ignored, CA Telon assumes ABEND processing, as follows:

- For all I/O except BROWSE, CA Telon invokes ABEND processing. OK is the default.
- For BROWSE, CA Telon does not invoke ABEND processing. ALL is the default.

IOAREA

The data area used for this I/O operation. The value in the field overrides a value in the COPYLBL field of another screen.

SEGMENT

The segment to which the SSALIST host variable applies. This value is informational only and does not affect code generation.

SSA LIST HOST VARIABLE

The SSA host variables used as the list of SSAs in the generated call.

Update VSAM or Sequential Detail Data Access

Access

On the Create/Update Data Group screen, enter **U** as a line command for a VSAM or SEQ data set.

Program ID

S146

Function

Updates I/O for a previously defined file.

XXXXXX.XX ** X-XXX-XXXXXXXX-XXXXXXX *****

COMMAND ==> _____

OPTIONS ==> PREVIEW __

GENERAL:

IGNORE _____

* IOAREA _____

* FDREC _____

VSAM:

KEY _____

* FUNC _____

* OPCODE _____

EXEC

CICS:

GENKEYL _____

* OPTLIST _____

* RECLTH _____

* READSET _____

Field Definitions

COMMAND

For information, see Primary Commands.

PREVIEW

A value to request a view of the generated call. Values include any nonblank character except a question mark.

IGNORE

Status codes to be ignored. All DBMS-specific ignore values are valid. These generic ignore values are translated to the corresponding DBMS-specific value at generation time:

Status Code	What CA Telon Ignores
OK	Blank or zero status codes
NOTFOUND or NFD	Not found status codes
DUPLICATE or DUP	Multiple record/key occurrences
LOGICERR or LOG	Errors on accesses that are dependent on prior required conditions
SECURITY or SEC	Security violations
ENDFILE or EOF	EOF conditions
NOTAVAIL or NAV	Conditions when resources are not available
NOSPACE	Out of space (sequential files only)
ALL	Any return code

If CA Telon encounters a return code (other than blanks) that is not included in the IGNORE field for an I/O, CA Telon assumes ABEND processing as follows:

- For all I/O except BROWSE, CA Telon invokes ABEND processing. OK is the default.
- For BROWSE I/O, CA Telon does not invoke ABEND processing. ALL is the default.

IOAREA

The data area used for this I/O operation. This value overrides a value in the COPYLV1 field specified on another screen.

FDREC

The name of the 01-level COBOL record name in the file definition (FD) that is written, if different from the default *dsname*-RECORD.

KEY

The names of the PL/I or COBOL variable containing the key to a record. If inheritance is requested, data access (user I/O) uses the variable name to identify a record that it retrieves. See the *Programming Concepts Guide* for information on inheritance.

FUNC

The nature of the call in conjunction with the value specified in the REQUEST field on the Create/Update Data Group screen. For example:

VSAM I/O Command	REQUEST Field	FUNC Field
DELETE	DELETE	DELETE
WRITE	CREATE	CREATE
REWRITE	REPLACE	REPLACE
READNEXT	READNEXT	READNEXT
ENDBR	READNEXT	ENDBR
READ	READ	READ
STARTBR1	READ	STARTBR
UNLOCK	READ	UNLOCK
READPREV	READ	READPREV
RESETBR	READ	RESETBR
ENDBR	READ	ENDBR
REWRITE	UPDATE	REWRITE
UNLOCK	UPDATE	UNLOCK

Note: ¹— To execute the equivalent of a STARTBR automatically through a program, the user must execute STARTBR and then reset the record position by entering a READPREV.

OPCODE

The operation code for VSAM access. Values are:

GTEQ

(Default.) Greater than or equal to.

EQ

Equal to

GT

Greater than

GENKEYL (VSAM Processing Only)

The length of the generic key used for the access.

Values include either an integer or the name of a COBOL or PL/I variable that contains the key-length value. If not specified here, CA Telon assumes that the access uses the full key length; it does not use the GENKEYL value from the RECORD statement.

OPTLIST (VSAM Processing Only)

Options on CICS data sets for command level calls. You can use this field to specify one or more of these options on the SEGMENT statement and all user exec data access (READ, UPDATE, CREATE, and DELETE). Separate each option with a comma. Values are:

- RRN
- SEGSET
- SEGSETALL
- SYSID
- MASSINSERT
- DEBKEY
- DEBREC
- UPDATE

Values coded on the SEGMENT statement are carried down to user exec specifications if you do not specify an OPTLIST value on the user exec data access.

Values that are not valid for a specific command level verb (such as READ, WRITE, and STARTBR) are automatically removed from any calls using that verb. For example, if MASSINSERT is specified as an option on the RECORD statement or user exec data access, it appears only if the EXEC CICS WRITE command is generated for that segment in the CICS program.

You can specify literal values for the SEGSET and SYSID options by enclosing the literal value in double quotes. For example:

```
OPTLIST=(RRN,SYSID("SYSA"))
```

RECLTH (VSAM Processing of Variable-length Records Only)

The maximum length of each record on the file. Values are:

Record-length

Maximum record length, specified as either an integer or the name of a COBOL or PL/I variable that contains the key-length value. This value is used when reading or writing the VSAM record. Any rewrite operations are processed using the current length of the record being updated (as determined by the read).

Read-length,rewrite-length

Maximum record length during a read and subsequent update, respectively. Each value can be specified as either an integer or the name of a COBOL or PL/I variable that contains the key-length value. Read-length is used for all automatic read EXEC CICS calls for the record. Rewrite-length is used as the maximum length of the updated record. This specification is applicable for UPDATE processing only (that is, usage is UPDATE).

The value in this field overrides the RECLTH value specified on the RECORD statement for the file being accessed.

Note: The length that you use in a CICS read operation for a variable-length record must be at least as large as the actual record retrieved. If not, a CICS abend results.

RECLTH (VSAM Processing of Variable-length Records Only)

The maximum length of each record on the file. Values are:

READSET

Generates a VSAM read for update with the SET option rather than INTO. When this parameter is set to Y, the Generator builds the EXEC CICS READ using the SET format. Values are as follows:

Y

Builds the EXEC CICS READ using the SET format.

N

Does not build the EXEC CICS READ using the SET format.

Update CICS Queue Record

Access

On the Create/Update Data Group screen, enter **U** as a line command for a CICS queue record.

Program ID

S13Q

Function

Updates CICS queue records.

HHNNNN.ND UPDATE CICS QUEUE RECORD ** *****

COMMAND=>

CQNAME

GENERAL:

TYPE

__ (TS/TD)

AUX/MAIN

(A/M)

*

LRECL

*

SYSID

*

LTHOPT

LTHLBL

*

QUELBL

RECORD:

LABEL

USAGE

*

COPY

*

COPYLV1

__ (Y/N)

COPYLBL

Field Definitions

COMMAND

For information, see Primary Commands.

CQNAME

The default queue name to identify this queue to CICS.

This field is protected if a value was specified in the NAME field on the Data Administration menu. If a value is displayed on this screen as a result of using the DGADD command on the Create/Update Data Group screen, you can modify the value here.

TYPE

(*Protected field.*) Identifies the type of CICS queue. Possible values are:

TS

Temporary storage

TD

Transient data

This value is the default for all program definitions that reference this queue.

AUX/MAIN (Temporary Storage Queues Only)

The type of storage to which this queue should be written. Values are:

A

(*Default.*) Auxiliary storage.

M

Main storage

This value is the default for all program definitions that reference this queue.

LRECL

The length (in bytes) of the queue record. If an LRECL value has been defined for the queue in data administration, you can override that value here.

This value is required if the LTHOPT value is also specified on this screen.

SYSID

The system ID to use when CA Telon accesses this queue.

A value is not required, but if you specify one, you must also specify LTHOPT and LRECL values on this screen.

This field is protected if a value was specified in the SYSID field on the Update CICS Queue Default Data screen. If a value is displayed on this screen as a result of using the DGADD command on the Create/Update Data Group screen, you can modify the value here.

The variable loaded with SYSID is on one of the following:

- SYSWK-*cqname*-QUEUE-SYSID (COBOL) or
- SYSWK_*cqname*_QUEUE_SYSID (PL/I), where *cqname* is the value of the CQNAME field. This is the default.

LTHOPT

A value to specify whether CA Telon is to generate the CICS LENGTH option in the I/O for the queue. Values are:

Y

Generate queue I/O with the CICS LENGTH option

N

(Default.) Generate queue I/O without the CICS LENGTH option

The default value is Y if you specified a value in the SYSID field or if this CICS queue has been defined in data administration; otherwise, it is N.

If you specify a value in the LRECL field and Y in the LTHOPT field, CA Telon uses the LRECL value to initialize the variable used with the LTHOPT option. This variable is one of the following:

- SYSWK-*cqname*-QUEUE-LENGTH (COBOL) or
SYSWK_*cqname*_QUEUE_LENGTH (PL/I), where *cqname* is the value of the CQNAME field. This is the default.
- If specified, the value in the LTHLBL on this screen

If you do not specify a value in the LRECL field, CA Telon-generated code does not initialize the CICS LENGTH option variable.

LTHLBL

The name of the COBOL or PL/I variable used with the LTHOPT value in I/O requests for this CICS queue. This value is meaningful only when the LTHOPT value is Y.

QUELBL

The name of the COBOL or PL/I variable that identifies the queue to be used in all EXEC CICS commands for this queue. The default value is:

- SYSWK-*cqname*-QUEUE-NAME (COBOL) or
SYSWK_*cqname*_QUEUE_NAME (PL/I), where *cqname* is the value in the CQNAME field.

LABEL

The label associated with this CICS queue. If the queue is defined in data administration, the default value is established. You can override it here.

This value replaces the CQNAME field value in the names that CA Telon generates for this queue.

USAGE

The type of processing associated with this CICS queue. Values are:

Request	Action
BROWSE	This queue can be used in an I/O loop by screen OUTPUT processing.
DEFINE	Explicitly request CA Telon to generate storage areas for this queue, regardless of whether data access has been defined for this item.
@DEFINE	CA Telon assumes that this item needs storage areas because this queue has data access defined to it.
DUMMY	This queue is not being used.
@DUMMY	CA Telon assumes that the queue is not being used because no data access has been defined for it; no storage areas are needed.

COPY

The COBOL COPY or PL/I INCLUDE member name that contains the queue record layout. If the queue is defined on the Update CICS Queue Default Data screen, a COPY value is inherited. You can override it on this screen.

If not specified, the program uses the CQNAME field value as the name of the member to copy or include.

If you enter **NONE**, no COPY or INCLUDE is generated.

COPYLV1

The starting level of the layout contained in the member identified in the COPY field. Values are:

Y

The COPY or INCLUDE layout for this queue starts at the 01 level. Specify also a COPYLBL field value to supply the variable name of the queue I/O area.

N

CA Telon generates the 01- and 02-level variables that precede the COPY or INCLUDE member.

If the queue is defined on the Update CICS Queue Default Data screen, a COPYLV1 value is inherited. You can override it on this screen.

COPYLBL

The name of the variable identifying the I/O area used in accessing this queue. If not specified, CA Telon generates all I/O for this queue using the data area IOA-*cqname*-QUEUE, where *cqname* is the value in the CQNAME field.

If the queue is defined on the Update CICS Queue Default Data screen, a COPYLBL value is inherited. You cannot override it on this screen.

Update CICS Journal Record

Access

On the Create/Update Data Group screen, enter **U** as a line command for a CICS journal record.

Program ID

S13J

Function

Updates CICS journal records.

HHNNNN.ND UPDATE CICS JOURNAL RECORD * *****			
COMMAND=> _____			
CJNAME _____			
GENERAL:	JFILEID	__	(01 - 99)
*	JTYPEID	__	
*	LRECL	__	
*	LTHOPT	__	(Y/N)
*	PFXLTH	__	
	LTHLBL	_____	
	PFXLBL	_____	
RECORD:	LABEL	_____	USAGE _____
*	COPY	_____	
*	COPYLV1	_____ (Y/N)	COPYLBL _____

Field Definitions**COMMAND**

For information, see Primary Commands.

CJNAME

(*Protected field.*) Displays the CA Telon name, inherited from the NAME field on the Data Administration menu, that identifies this CICS journal definition.

JFILEID

The CICS journal ID in all generated I/O for this journal. Valid CICS journal IDs are 02 through 99. (CICS reserves 01 for the system log.)

If this journal has been defined in data administration, this field is protected. If the journal was defined on the Create/Update Data Group screen, you can modify the JFILEID value here.

JTYPEID

A 2-character identifier of the origin of the journal record.

A value is required in this field. If the journal has been defined in data administration, this field is protected. If the journal was defined on the Create/Update Data Group screen, you can modify the JTYPEID value here.

LRECL

The maximum length (in bytes) of the user data of this journal's entries.

A value is required in this field. If the journal has been defined in data administration, you can accept the default value or modify it here. The value of this field can be defaulted if this journal exists.

LTHOPT

A value to specify whether CA Telon is to generate the CICS LENGTH option in the I/O for this journal. Values are:

Y

Generate journal I/O with the CICS LENGTH option

N

(Default.) Generate journal I/O without the CICS LENGTH option.

LTHLBL

The name of the COBOL or PL/I variable containing the length in I/O requests for this CICS journal.

If Y is specified for LTHOPT, the default value for LTHLBL is
SYSWK-*cjname*-JOURNAL-LENGTH (COBOL) or
SYSWK_*cjname*_JOURNAL_LENGTH (PL/I), where *cjname* is the value in the CJNAME field.

This value is invalid if LENGTH is not specified for a journal data access request.

PFXLTH

The length of user prefix data included in this journal I/O request. If not specified, CA Telon does not generate the PREFIX option.

PFXLBL

The name of the COBOL or PL/I variable containing the prefix for this journal I/O request.

If a PFXLTH value without a PFXLBL value is specified, CA Telon generates the default variable `SYSWK-cjname-JOURNAL-PREFIX` (COBOL) or `SYSWK_cjname_JOURNAL_PREFIX` (PL/I), where *cjname* is the value in the JOURNAL field.

PFXLBL is invalid if no value is specified in the PFXLTH field.

LABEL

The label that CA Telon uses in generating variable and procedural names for the journal.

The default value is the value in the CJNAME field. You can modify it here.

USAGE

The type of processing associated with this CICS journal. Values are:

Request	Action
DEFINE	Explicitly request CA Telon to generate storage areas for this journal, regardless of whether data access has been defined for this item.
@DEFINE	CA Telon assumes that this item needs storage areas because this journal has data access defined to it.
DUMMY	This journal is not being used.
@DUMMY	CA Telon assumes that the journal is not being used because no data access has been defined for it; no storage areas are needed

COPY

The name of the COPY or INCLUDE member that contains the journal record layout.

If not specified, the default is the value in the CJNAME field.

If you enter **NONE**, CA Telon does not generate a COPY or INCLUDE for this journal.

COPYLV1

The starting level of the layout contained in the member identified in the COPY field. Values are:

Y

The COPY or INCLUDE layout for this journal starts at the 01 level. Specify also a COPYLBL field value to supply the variable name of the queue I/O area.

N

CA Telon generates the 01- and 02-level variables that precede the COPY or INCLUDE member.

If the journal is defined on the Update CICS Queue Default Data screen, a COPYLV1 value is inherited. You can override it on this screen.

COPYLBL

The name COBOL or PL/I group-level variable that identifies the I/O area used in accessing the journal. For example, if the data area for this journal begins with the COBOL variable 01 JOURNAL-SAVE, the COPYLBL value must be JOURNAL-SAVE.

If the journal is defined in data administration, a default value is displayed here. You can modify it.

Update CICS Queue Data Access

Access

On the Create/Update Data Group screen, enter **U** as a line command for a CICS queue I/O request.

Program ID

S14Q

Function

Supports queue I/O requests.

XXXXXX.SD ** U-100-READ-TRGQUEUE **** *****

COMMAND ==>

OPTIONS ==> PREVIEW

GENERAL

TYPE TS

* IGNORE

* IOAREA

EXEC

CICS:

* SYSID

* LTHOPT @Y/N

* SET Y/N

* NOSUSP Y/N

* ITEM Y/N

* NUMITEM Y/N

LTHLBL @

SETLBL

ITMLBL

NITMLBL

Field Definitions

COMMAND

For information, see Primary Commands.

PREVIEW

A value to request a view of the generated call. Values include any nonblank character except a question mark.

TYPE

(Protected field.) Identifies the type of CICS queue. Possible values are:

TS

Temporary storage

TD

Transient data

AUX/MAIN (Temporary Storage Queues Only)

(*Protected field.*) Identifies the type of storage to which the queue is written. Possible values are:

A

(*Default.*) Auxiliary storage.

M

Main storage

IGNORE

Conditions that the program is to ignore for this call. You can specify any of the valid CICS queue handle conditions and these generic status codes:

Status Code	What CA Telon Ignores
OK	Successful journal access
NOTFOUND or NFD	ITEMERR (only valid for temporary storage access for which the ITEM option has been specified)
ALL	All conditions including OK

CA Telon translates generic codes into the correct CICS queue conditions at program generation.

The default condition is OK for all queue I/O requests except BROWSE (for CICS screen programs) and TRANSACT (for CICS nonterminal). The default is ALL for BROWSE access requests.

If any condition not specified is encountered during processing, CA Telon-generated codes invoke abnormal-termination processing.

Valid CICS handle conditions for temporary storage queues are:

- INVREQ
- IOERR
- ISCINVREQ
- ITEMERR
- LENGERR
- NOSPACE
- NOTAUTH
- QIDERR
- SYSIDERR

Valid CICS HANDLE conditions for transient data queues are:

- IOERR
- ISCINVREQ
- LENGERR
- NOSPACE
- NOTAUTH
- NOTOPEN
- QBUSY
- QIDERR
- QZERO
- SYSIDERR
- DISABLED

IOAREA

The data area used for the I/O operation. This value overrides the COPYLV1 field value on the Update CICS Queue Record screen.

IOAREA is inherited.

SYSID

(Protected field.) Identifies the system ID used when accessing the queue. If a value is displayed, also enter a value in the LTHOPT field.

LTHOPT

A value to specify whether CA Telon is to generate the LENGTH option in queue I/O. Values are:

Y

Generate queue I/O with the CICS LENGTH option

N

Generate queue I/O without the CICS LENGTH option

The default is Y for read and write requests; otherwise, it is N.

If you specify a value in the LRECL field and Y in the LTHOPT field, CA Telon uses the LRECL value to initialize the variable used with the LTHOPT option. This variable is one of the following:

- SYSWK-*cqname*-QUEUE-LENGTH (COBOL) or SYSWK_*cqname*_QUEUE_LENGTH (PL/I), where *cqname* is the value of the CQNAME field. This is the default.
- If specified, the value in the LTHLBL on this screen

If you do not specify a value in the LRECL field, CA Telon-generated code does not initialize the CICS LENGTH option variable.

For queues, LTHOPT is equal to Y and protected temporary storage.

LTHLBL

The name of the COBOL or PL/I variable used with the LTHOPT value in I/O requests for this CICS queue. This value is inherited from the Update CICS Queue Record screen. It is meaningful only when the LTHOPT value is Y.

SET

A value to specify whether the SET option is generated in queue I/O. Values are:

Y

Generate queue I/O with the SET option

N

(hp1.Default.) Generate queue I/O without the SET option.

SETLBL

The name of the COBOL or PL/I variable used when the SET option is requested in I/O for the queue (that is, when the value in the SET field is Y).

The default is SYSWK-*cqname*-QUEUE-PNTR (COBOL) or SYSWK_*cqname*_QUEUE_PNTR (PL/I), where *cqname* is the value in the CQNAME field on the Update CICS Queue Record screen.

NOSUSP

A value to specify whether these conditions are ignored:

- NOSPACE on a TS WRITE I/O request
- QBUSY on a TD READQ request

The value in this field applies to only these two I/O requests and is ignored if specified for any others. It is not meaningful for CICS version 1.6 or earlier, where NOSUSPEND is not a valid parameter.

Values are:

Y Ignore the condition

N Do not ignore the condition

ITEM

A value to specify whether the ITEM option is generated in the I/O for this queue. It is valid for temporary storage queues only; the TYPE field value must be TS. Values are:

Y Generate queue I/O with the CICS ITEM option

N (Default.) Generate queue I/O without the CICS ITEM option.

ITEMLBL

The name of the COBOL or PL/I variable used when the ITEM option is requested in the I/O for the queue (that is, when the value in the ITEM field is Y). It is valid for temporary storage queues only; the TYPE field value must be TS.

The default is SYSWK-cqname-QUEUE-ITEM (COBOL) or SYSWK_cqname_ITEM (PL/I), where cqname is the value in the CQNAME field on the Update CICS Queue Record screen.

NUMITEM

A value to specify whether the NUMITEM option is generated in queue I/O, to indicate how many items are in the queue. It is valid for only a TS READQ I/O request. Values are:

Y Generate queue I/O with the CICS NUMITEM option

N (Default.) Generate queue I/O without the CICS NUMITEM option.

NITMLBL

The name of the COBOL or PL/I NUMITEM variable used when the NUMITEM option is requested in I/O for this queue (that is, when the value in the NUMITEM field is Y). It is valid for temporary storage queues only; the TYPE field value must be TS.

The default is SYSWK-cqname-QUEUE-NITEM (COBOL) or SYSWK_cqname_QUEUE_NITEM (PL/I), where cqname is the value in the CQNAME field on the Update CICS Queue Record screen.

Update CICS Journal Data Access

Access

On the Create/Update Data Group screen, enter **U** as a line command for a CICS journal I/O request.

Program ID

S14J

Function

Supports the journal I/O request, JOURNAL.

```

XXXXXX.SD ** U-100-JOURNAL-JJJJJJJJ *** *****
COMMAND=>
OPTIONS => PREVIEW
JOURNAL => JFILEID ____ JTYPEID __

GENERAL: IGNORE _____
*         IOAREA @_____

EXEC
CICS:    FUNC
*        LTHOPT ____ (Y/N)      LTHLBL @_____
*        PFXLTH @____          PFXLBL @_____
*        REQID  ____ (Y/N)      REQDLBL _____
*        WAIT   ____ (Y/N)
*        STARTIO ____ (Y/N)
*        NOSUSP ____ (Y/N)

```

Field Definitions

COMMAND

For information, see Primary Commands.

PREVIEW

A value to request a view of the generated call. Values include any nonblank character except a question mark.

JFILEID

(*Protected field.*) Displays the CICS journal ID, used in all generated I/O for this journal.

JTYPEID

(*Protected field.*) Displays the identifier of the origin of the journal record.

IGNORE

Conditions that the program is to ignore for this call. You can specify any of the valid CICS queue handle conditions and these generic status codes:

Status Code	What CA Telon Ignores
OK	Successful journal access
NOTFOUND or NFD	ITEMERR (only valid for temporary storage access for which the ITEM option has been specified)
ALL	All conditions including OK

CA Telon translates generic codes into the correct CICS queue conditions at program generation.

The default condition is OK for all queue I/O requests except BROWSE (for CICS screen programs) and TRANSACT (for CICS nonterminal). The default is ALL for BROWSE access requests.

See Update CICS Queue Data Access for valid CICS handle conditions. If any condition not specified is encountered during processing, CA Telon-generated codes invoke abnormal-termination processing.

IOAREA

The data area used for the I/O operation. This value overrides the COPYLV1 field value on the Update CICS Journal Record screen.

IOAREA is inherited.

FUNC

The nature of the journal I/O request. Values are:

WAIT

CA Telon generates an EXEC CICS WAIT JOURNAL instruction. The options WAIT, NOSUSP, PFXLTH, and PFXLBL cannot be used and should not be specified on this screen. Only STARTIO and REQID are valid.

(blank)

CA Telon generates an EXEC CICS JOURNAL instruction.

LTHOPT

A value to specify whether CA Telon is to generate the LENGTH option in queue I/O. Values are:

Y

Generate queue I/O with the CICS LENGTH option

N

Generate queue I/O without the CICS LENGTH option

LTHLBL

The name of the COBOL or PL/I variable that contains the length option in I/O requests for the journal.

The default is SYSWK-*cjname*-JOURNAL-LENGTH (COBOL) or SYSWK_*cjname*_JOURNAL_LENGTH (PL/I), where *cjname* is the value in the CJNAME field on the Update CICS Journal Record screen.

A value in this field is *not* valid under either of these conditions:

- The value in the LTHOPT field is N
- The value in the FUNC field is WAIT

PFXLTH

The length of user prefix data included in the journal I/O request. If not specified, the PREFIX option is not generated.

If the value in the FUNC field is WAIT, a value is not valid and causes a CA Telon generation error.

PFXLBL

The name of the COBOL or PL/I containing the prefix for the journal I/O request. The default is SYSWK-*cjname*-JOURNAL-PREFIX (COBOL) or SYSWK_*cjname*_JOURNAL_PREFIX (PL/I), where *cjname* is the value in the JOURNAL field on the Update CICS Journal Record screen. A value in this field is *not* valid under either of these conditions:

- The value in the PFXLTH field is blank
- The value in the FUNC field is WAIT

REQID

A value to specify whether asynchronous output is required. Values are:

Y

Asynchronous output is required

N

Asynchronous output is not required

REQIDLBL

The name of the COBOL or PL/I variable that contains the REQID in I/O requests for the journal.

The default is SYSWK-*cjname*-JOURNAL-REQID (COBOL) or SYSWK_*cjname*_JOURNAL_REQID (PL/I), where *cjname* is the value of the CJNAME field on the Update CICS Journal Record screen.

A value in this field is valid only if you specify a value in the REQID field.

WAIT

A value to specify whether the I/O requests for the journal are to include a WAIT option, to cause a program containing the I/O request to wait until it is notified of the completion of the request. Values are:

Y

Include the WAIT option

N

(*Default.*) Do not include the WAIT option.

STARTIO

A value to specify whether the output of the journal record is initiated immediately. Values are:

Y

Initiate the output immediately

N

(*Default.*) Do not initiate the output immediately.

NOSUSP

A value to specify whether the NOJBUFSP condition is ignored by a JOURNAL I/O request. Values are:

Y

Ignore the condition

N

Do not ignore the condition

A value in this field is ignored if the value in the WAIT field is Y. It is not meaningful for CICS version 1.6 or earlier, where NOSUSPEND is not a valid parameter.

Update TPPCB Detail Data Access

Access

On the Create/Update Data Group screen, enter **U** as a line command for a TPPCB.

Program ID

S149

Function

Updates I/O for the teleprocessing PCB defined on the Create/Update Data Group screen.

```
XXXXX.XX  ** X-XXX-XXXXXXXX-XXXXXXXX *****  
COMMAND ==> _____  
  
GENERAL:  FUNC    _____  
  
I/O:      TPPARMS _____  
*          * _____  
*          * _____  
*          * _____
```

Field Definitions

COMMAND

For information, see Primary Commands

FUNC

The nature of the call, in conjunction with the value specified in the REQUEST field on the Create/Update Data Group screen. Values are:

- GU
- GN
- ISRT
- CHNG
- PURG
- CMG
- GCMD
- XRST
- SYNC
- SNAP
- CHKP
- DEQ
- GSCD
- LOG
- ROLB
- ROLL
- STAT
- SYMCHKP

TPPARMS

The parameter list passed to a teleprocessing PCB with the WRITE request. Separate each parameter with a comma.

The first parameter in the list (that is, the first parameter after the PCB), represents the third parameter in the call.

TPPARMS and IOAREA are mutually exclusive.

If DBMS is equal to EXEC DLI and you are coding either CHKP, XRST, or SYMCHKP in a batch program, use the parameters in this table as indicated:

Parameter	Usage
CHKP	ID(PARM1)
XRST	ID(PARM1)
	MAXLENGTH(PARM2)
	AREA1(PARM3)
	LENGTH1(PARM4)
	.
	.
	.
SYMCHKP	AREA7(PARM15)
	LENGTH7(PARM16)
	ID(PARM1)
	AREA1(PARM2)
	LENGTH(PARM3)
	.
	.
	AREA7(PARM14)
	LENGTH(PARM15)

List/Show Custom Code

Access

On the Online Program Definition menu, enter:

- **LI** in the FUNCTION field
- **CC** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field
- **SD, DR, or RD**, in the TYPE field
- *Custom-code* in the CUSTCODE field

On the Nonterminal Program Definition menu, enter:

- **LI** in the FUNCTION field
- **CC** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field
- **ND** in the TYPE field
- *Custom-code* in the CUSTCODE field

You can also access this screen by selecting the custom code option on these screens:

- Create/Update Screen Definition
- Create/Update Driver Definitions
- Create/Update IMS/DC Report Definitions
- Create/Update Nonterminal Definition
- Create/Update Batch Definitions

Program ID

S159

Function

Lists custom code members.

```
XXXXXX.SD LIST CUSTOM CODE *****
COMMAND ==>                                     PAGE 01

**NAME** *RENAME* *****DESCRIPTION***** **USER**  UPDATED
TESTCC01          TEST CUSTOM CODE MEMBER          FOX007  08/19/05
```

Field Definitions

COMMAND

For information, see Primary Commands.

Note: You can specify the starting point of the list by entering **L member-name**. If **member-name** is found, the list begins with **member-name**. If **member-name** is not found, the list begins with the name of the member that follows *member-name* in alphabetical order.

(Item manipulation)

A field allowing entry of a control character to manipulate the corresponding item on the screen:

Entry	Function	Description
C	COPY	Copies a member
D	DESCRIPTION	Changes the description of a member
R	RENAME	Renames a member

Entry	Function	Description
S	SHOW	Allows you to enter a member in browse mode
U	UPDATE	Allows you to enter a member in update mode
Z	ZAP	Purges a member without a confirmation
P	PURGE	Deletes a member after confirmation

NAME

The name of the custom code member.

RENAME

The name of the member to be renamed or copied, if the value in the item manipulation field is R or C.

Note: After a member has been renamed, it is accessed under its new name.

After processing an item manipulation function, CA Telon might display a confirmation message in this field:

Function Entry	Confirmation Message
C	*COPIED
D	*DESC UP
P	None
R	*RENAMED
S	None
U	*PROCSD
Z	*PURGED

DESCRIPTION

A description of the custom code member.

USER

(*Protected field.*) Identifies user who last accessed the custom code.

UPDATED

(*Protected field.*) Identifies when the custom code was last accessed.

Edit Custom Code

Access

On the Online Program Definition menu or the Nonterminal Program Definition menu, enter:

- **CR** or **UP** in the FUNCTION field
- **CC** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field
- *Custom-code* in the CUSTCODE field

You can also access this screen by entering **U** in the item manipulation field for the custom code member on the List/Show Custom Code screen.

You can view, but not edit, the custom code member by invoking preview processing from one of these screens:

- Update SQL Detail Data Access
- Update DL/I Detail Data Access
- Update Database User I/O SSALIST
- Update VSAM or Sequential Detail Data Access
- Update CICS Queue Data Access
- Update CICS Journal Data Access
- Update TPPCB Detail Data Access

Program ID

S151

R	Repeat a line
RR	Repeat a block of lines
O	Overlay a line
OO	Overlay a block of lines
A	Line(s) to insert, copy, or move go after this line
B	Line(s) to insert, copy, or move go before this line
D	Delete an entry
X(<i>n</i>)	Exclude <i>n</i> lines beginning with this line
XX	Exclude a block of lines

See Line Commands for more information.

(Edit line) A

An input line for custom code. You can enter code as you would on an SPF editor.

Note: A custom code member may not exceed 10,000 lines. Computer Associates strongly recommends that no custom code member exceed 2000 lines, as unpredictable results may occur.

Update TSO or IMS Screen Environment

Access

On the Online Program Definition menu, enter:

- **CR** or **UP** in the FUNCTION field
- **EN** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field
- **SD** in the TYPE field
- **IMS** in the ENVIRON field

You can also access this screen by selecting the ENV IMS option.

Program ID

S162

Function

Specifies the environment characteristics of a CA Telon generated program that operates in an IMS or CA Telon TSO test environment. These characteristics must be specified before you can generate a program to run under IMS.

```

HEADERID.SD UPDATE TSO/IMS SCREEN ENV * *****
COMMAND ==> _____

IMS:  LINKOPT (D/S)  CONVERS (Y/N)  LINEOPT (1/2/3)
*    TRACE  (Y/N)   PGM/PSB NAME
*    GENPCBS (Y/N)   LNKCOPY _____ USGCOPY _____
TSO:  GENPCBS (Y/N)   LNKCOPY _____ USGCOPY _____
TRAN: TRANCDE _____ TRANMFS (Y/N)  TRANFLD _____
SPA:  SPASIZE _____ SPACMPT (Y/N)
*    WKSPASZ _____ WKSPAI0: GET _____ PUT _____ WKSPAIN (Y/N)

FLOW: LINKPGM _____
*    (ID"S) _____
*    _____
*    MSGPGM _____
*    (ID"S) _____
*    _____
*    MSGTRAN _____
*    (ID"S, _____
*    PGM"S) _____
*    MSGTBL _____ MSGBUF _____

MSG ERR: A4EPMG _____ A4EMSG _____
MFS:  MFSMOD _____ SYMSMG _____ MIDONLY/DEVICE
PLIXOPT: _____ IMS _ TSO _ ALL _ (C-CREATE/U-UPDATE/P-PURGE)

```

Show/Purge screen

You can access the Show/Purge TSO or IMS Screen Environment (S161) screen from the Online Program Definition menu by entering:

- **SH** or **PU** in the FUNCTION field
- **EN** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field
- **IMS** in the ENVIRON field

Field Definitions

The Show/Purge TSO or IMS Screen Environment screen fields are the same as the Update TSO or IMS Screen Environment screen fields.

COMMAND

For information, see Primary Commands.

LINKOPT

The type of linking to occur in the IMS program. Values are:

D

The program is a stand-alone program that dynamically links to other programs. If you do not specify ANY in either the LINKPGM or MSGPGM fields, all programs to which control can be passed must be accounted for in MSGTRAN, MSGPGM, MSGTBL, or LINKPGM values. Any attempt to transfer to programs not identified in these fields results in a user ABEND.

S

Generate a program as a subroutine that executes under a driver. Values in the CONVERS, MFSMOD, and PGMNAME are valid with this option.

The default value is set at installation.

CONVERS

A value to specify whether the system that CA Telon generates is IMS conversational. Values are:

Y

The system being generated is an IMS conversational system using an IMS SPA

N

The system being generated is an IMS non-conversational system using a WORKSPA database

In **conversational** transaction processing, IMS maintains a conversation with the terminal through an IMS SPA. While in conversation with IMS, the transaction code is maintained by IMS until modified by the application program through message switching or setting to spaces to terminate the conversation.

In **non-conversational** processing, the transaction code is sent to IMS each time Enter or any PF key is pressed by the application user. When you use the non-conversation mode, CA Telon maintains a pseudo-conversation with IMS through the use of a WORKSPA database. This is done by using the SPA-TRANSACTION-CODE field in the transfer work area in the same way that IMS uses this field in the IMS SPA.

LINEOPT

The line optimization logic that the program uses and generates. Values are:

1

Use the CA Telon line optimization subroutine. CA Telon automatically performs line optimization for you.

2

Simulate subroutine optimization in procedural custom code (except for fill-character processing).

3

Do not generate the line optimizing code.

Note: For PWS, the LINEOPT value must be 2 or 3.

TRACE

A value to specify whether the program generates and maintains TRACE variables for debugging. Values are:

Y

The program generates TRACE variables.

N

The program does not generate TRACE variables.

Note: TRACE variables increase the size of the generated program. TheTRACE should be N for production environment.

PGM/PSB NAME

The name of the load module that the linkage editor creates. CA Telon requires this field only when the load module name is different from the name that CA Telon generates for the program. The generated program name is set as an installation default.

GENPCBS

A value to specify whether to include DL/I PCB masks in the program. Values are:

Y

Automatically generate PCB masks in the program

N

PCB masks must be included in the LNKCOPY and USGCOPY members

LNKCOPY

The name of the COPY/INCLUDE member containing the 01-level declarations included in the linkage section.

USGCOPY

The variable declarations in the linkage member appended to the COBOL procedure division or the PL/I procedure statements.

In COBOL, the variable declaration is the list of 01-level variables in the linkage section. In PL/I, variable declaration is a list of DECLARE statements in the linkage section.

GENPCBS

A value to specify whether to include TSO PCB masks in the program. Values are:

Y

Automatically generate PCB masks in the program

N

PCB masks must be included in the LNKCOPY and USGCOPY members

LNKCOPY

The name of the COPY/INCLUDE member containing the 01-level declarations included in the linkage section.

USGCOPY

The variable declarations in the linkage member appended to the COBOL procedure division or the PL/I procedure statements.

In COBOL, the variable declaration is the list of 01-level variables in the linkage section. In PL/I, variable declaration is a list of DECLARE statements in the linkage section.

TRANCDE

The IMS transaction code associated with the generated program. Enter a value only when the IMS transaction is different from the CA Telon-generated program name. The program name is set as an installation default.

TRANMFS

A value to specify whether the MFS processes the transaction code. Values are:

Y

MFS processes the transaction code

N

(Default.) MFS does not process the transaction code.

TRANFLD

The transaction code that is imbedded to the MID for this screen. This value is used only when the /FORMAT command is used to start the application, or if the system is non-conversational.

SPASIZE

The size of the SPA specified in the IMS generation for this application. It can be defined alone or in conjunction with the WKSPASZ field value. The total of the values in the two fields must be enough bytes to hold:

- SPA header
- Application transfer work area
- Size of the largest screen image in the application

You can fine tune the size as needed, based on SPA requirements printed out on generated program listings.

SPACMPT

A value to specify whether CA Telon is to generate a SPA with a fixed number of overhead bytes, making it compatible for use by static and dynamic programs. The generation of such a SPA allows for message switching between static and dynamic modules in an application. Values are:

Y

CA Telon generates a compatible SPA. CA Telon generates a field called SPACMPAT in the dynamic programs where the next-program-name field exists in static programs.

N

CA Telon generates SPA with a different number of overhead bytes, based on whether the module is static or dynamic. Message switching between static and dynamic modules cannot take place in the application.

WKSPASZ

The number of bytes in the WORKSPA database used in this application system. The value is not used when you do not use a WORKSPA database.

In a non-conversational system, the database must be large enough to hold:

- The application transfer work area
- Overhead
- The largest screen image in the application

In a conversational system, the WKSPASZ value must specify enough bytes to hold the overflow from the IMS SPA area, whose size is specified in the SPASIZE field.

WKSPAIO

The name of two copy members containing custom code used in the retrieval and writing of WORKSPA databases:

- **GET** contains custom code included before the read of the WORKSPA database
- **PUT** contains custom code included before the replace of the WORKSPA database

WKSPAIN

A value to specify whether the generation of CA Telon WORKSPA database initialization code is to take place in IMS program section C-920-GET-WORKSPA. Values are:

Y

Generate code to reinitialize the CA Telon transfer work area when the IMS program flow has been broken

N

Do not generate such code

LINKPGM

The program IDs of all programs to which the generated program can be dynamically linked through CA Telon, specified in this format:

id[,id.]

where:

id

The CA Telon-generated program ID in the *hhnnnn* format of your CA Telon installation.

The list of IDs may be a maximum of 253 bytes long.

Alternately, you can enter **ANY** to specify the generated program is dynamically linked by means of CA Telon to NEXT-PROGRAM-NAME if NEXT-PROGRAM-NAME-ID is not detected in a search of the MSGPGM or MSGTRAN fields. If you enter **ANY** here, do not specify ANY in the MSGPGM field or any value in the MSGTBL field.

MSGPGM

The program IDs of all programs to which this program issues IMS message switches to transfer control, specified in this format:

id[,id.]

where:

id

The CA Telon-generated program ID in the *hhnnnn* format of your CA Telon installation.

The list of IDs may be a maximum of 253 bytes long.

Alternately, you can enter **ANY** to specify that the generated program is to do a message switch to NEXT-PROGRAM-NAME if the NEXT-PROGRAM-NAME-ID is not detected in LINKPGM, MSGTRAN, or MSGTBL specifications. If you enter **ANY** here, do not specify ANY for LINKPGM.

This value is valid if the transaction code for the program to receive control is the CA Telon-generated transaction code. Otherwise, specify a value in the MSGTRAN field.

MSGTRAN

The pairing of screen program IDs and corresponding IMS transaction codes, specified in this format:

id,tran[,id,tran.]

where:

id

The CA Telon-generated program ID in the *hhnnnn* format of your CA Telon installation.

tran

The eight-character IMS transaction code equated with *id*

This field is limited to 256 characters. To exceed this limit, use the MSGTBL field instead.

The generated program can transfer control by issuing IMS message switches. Specify values only if the transaction code of the program to receive control is different from the CA Telon-generated transaction code.

MSGTBL

The copy member containing the list that would otherwise be entered in the MSGTRAN field.

MSGBUF

The member name and length of a user-defined buffer area for use in automatic message switching. CA Telon appends the member name to the definition for the TP-OUTPUT-XFER-BUFFER.

A4EPGM

A program ID processed as the next program when an A4 status code results from attempting an IMS message switch. The program ID must also be defined in the LINKPGM, MSGPGM, MSGTRAN, or MSGTBL field.

This field and A4MSG are mutually exclusive.

A4MSG

The message displayed in the ERRMSG1 field when an A4 status results from attempting an IMS message switch.

Values include literal message text or the name of the variable containing the message. Enclose literal message text in single quotes.

This field and the A4EPGM field are mutually exclusive. This field is invalid when you specify S in the LINKOPT field.

The value supplied in the A4MSG parameter is only referenced in the C-400-TERMIO-XFER-MSG-SWITCH section. That section is only generated if LINKOPT=D and if one of the following MESSAGE-SWITCH parameters is specified: MSGPGM, MSGTRAN, or MSGTBL. If one of these parameters is entered, the above section is generated and contains the MOVE statement which assigns the value in the A4MSG parm to TPO-ERRMSG1.

MFSMOD

The MFS MOD name generated for this program if it is different from the CA Telon-generated MOD name. You can use this value to create a meaningful MOD name for users when the FORMAT command is used to start the application.

SYSMSG

The name of the field in the CA Telon-generated program to which system messages are sent. Use this field only when the program field designated for the system messages is not the special CA Telon field SYSMSG (that is, if the SYSMSG field is not part of the panel definition).

For example, for the system to send its messages to the field defined as ERRMSG1, enter **ERRMSG1** here.

MIDONLY / DEVICE

A field that enables you to transfer to the Update IMS MFSs screen to create, update, or purge MIDONLY fields in the screen definition. Values are:

C

Create a MIDONLY statement for this environment

P

Purge a MIDONLY statement from this environment

U

Update a MIDONLY statement for this environment

Note: To update, you can enter any nonblank value other than C or P.

PLIXOPT

(Displayed only when language is PL/I.) A field that enables you to transfer to the Update PL/I Executable Options screen to create, update, or purge PL/I executable options that override installation-defined PL/I defaults for CA Telon-generated programs. Values are:

C

Create a PLIXOPT statement for this environment

P

Purge a PLIXOPT statement from this environment

U

Update a PLIXOPT statement for this environment

Update IMS MFSs

Access

On the Update TSO or IMS Screen Environment screen, enter any nonblank value in the MIDONLY field.

Program ID

S163

TYPE

The device type to override the default device type. Values are:

Definition Type	Device Type	Size
Screen (SD)	3270, 2 (Default)	24 x 80
	3270, 1	12 x 80
	3270-A3	32 x 80
	3270-A4	43 x 80
	3270-A7	27 x 132
Report (RD)	3270P, 2 (Default)	55 x 120

FEAT

The features for this device or program group. Device features include:

- Print line 120
- Print line 126
- Print line 132
- Data entry keyboard
- Program function keys
- Selector light pen detect
- Operator identification card reader
- Dual platen
- User-defined features for the SCS1 and SCS2 devices and DPM programs.

Alternately, the default value is IGNORE, which specifies the device features are ignored for this device.

EATTR

A value to specify whether to use extended attributes on the screen. Values are:

Y

Use extended attributes

N

(Default.) Do not use extended attributes.

LTH

The length of the MIDONLY field.

MIDSOURCE

The name of the field added to the beginning of the MID but not to appear on the screen. This value is used when Y is specified in the FMTCNTL field in the panel definition.

See Update Literal Fields or Update Output/Input/Outin Field for more information on the FMTCNTL field.

LABEL

The CA Telon name for the MIDONLY field.

INITIALIZATION

The value to which CA Telon initializes the MIDONLY field.

Update PL/I Executable Options

Access

On the Update TSO or IMS Screen Environment screen or the Update CICS Screen Environment screen, enter the appropriate character in the PLIXOPT field.

Program ID

S164

Function

Generates the PLIXOPT statement.

HEADERID.SD UPDATE PL/I EXECUTION OPTS *****

COMMAND ==> _____

ENTER "PURGE" CR "PPPP" TO CONFIRM PURGE REQUEST

ENVIRONMENT

ALIGN

A-ALIGNED

U-UNALIGNED

STORAGE

S-STATIC

A-AUTOMATIC

REORDER

(Y/N)

XOPTS

About the PLIXOPT statement

For the programs that CA Telon generates, a PLIXOPT statement can override installation-defined PL/I defaults for:

- Execution options
- Storage options
- Variable alignments options
- Reorder compiler options

CA Telon inserts the PLIXOPT after the first statement in the definition (the SCREEN, REPORT, or DRIVER statement) and prior to the program generation statement (the CICSPGM, IMSPGM, IMSDRV, or TSOPGM statement). On this screen you can code multiple PLIXOPT statements to define options for different environments.

Field Definitions

COMMAND

For information, see Primary Commands.

ENVIRONMENT

The environment. This value is inherited from the screen from which you transferred.

ALIGN

A value to specify whether CA Telon is to align the program variables. Values are:

A

(Aligned) Align variables

U

(Unaligned) Do not align variables

STORAGE

A value to specify whether CA Telon is to allocate variable storage at execution time. Values are:

S

(Static) Do not allocate variable storage at execution time

A

(Automatic) Allocate variable storage at execution time

REORDER

A value to specify whether variable storage is to reorder the program for efficiency reasons. Values are:

Y

Reorder the program

N

Do not reorder the program

XOPTS

The execution options for the PL/I CA Telon program. Option specifications must be delimited by a comma.

Update CICS Screen Environment

Access

To access the CICS screen definition view of this update screen, on the Create/Update Screen Definition screen enter a nonblank character to select the ENV field when the value is CICS.

To access the nonterminal definition view of this update screen, on the Create/Update Nonterminal Definition screen enter a nonblank character to select the ENV field when the value is CICS.

You can access a view of this screen also by entering the following on the Online Program Definition menu or the Nonterminal Program Definition menu:

- **CR, UP, or SH** in the FUNCTION field
- **EN** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field
- **CICS** in the ENVIRON field

Program ID

S165

Function

Specifies the CICS environment data for the screen definition or nonterminal definition, including the characteristics of the CICS program and the BMS control blocks.

CICS screen definition view

```

HHHHH.SD UPDATE CICS ENVIRONMENT *** *****
COMMAND ==> _____

GENERAL: DBMS      _____  TRACE  (Y/N)  LINEOPT  (1/2/3)
*        SPASIZE   _____  TRANCODE _____
*        LNKCOPY   _____  USGCOPY  _____A

BMS : BMS      (Y/N)  BSMAP   _____

STORAGE: SPASTG   (A/S)  IOASTG  (A/S)  TPBSTG  (A/S)

DL/I  : PSBSCHD  (Y/N)  PSBNAME _____
*      GENPCBS  (Y/N)

PL/I  : PLIXOPT  (C-CREATE/U-UPDATE/P-PURGE)

```

CICS Nonterminal definition view

```

HHNNNN.ND UPDATE CICS ENVIRONMENT *** *****
COMMAND ==> 1_____

GENERAL: DBMS      _____  TRACE  (Y/N)
*        SPASIZE   _____  TRANCODE _____
*        LNKCOPY   _____  USGCOPY  _____

STORAGE: SPASTG   (A/S)  IOASTG  (A/S)

DL/I  : PSBSCHD  (Y/N)  PSBNAME _____
*      GENPCBS  (Y/N)

PL/I  : PLIXOPT  (C-CREATE/U-UPDATE/P-PURGE)

```

Show/Purge screen

You can access the Show/Purge CICS Screen Environment (S166) screen from the Online Definition menu screen by entering:

- **SH** or **PU** in the FUNCTION field
- **FG** or **PS** in the ITEM field
- *Name* in the NAME field
- *Identifier* in the ID field

Alternately, on the Show/Purge Nonterminal Definition screen, you can enter a nonblank character to select ENV CICS.

Field Definitions

The Show/Purge CICS Screen Environment screen fields are the same as the Update CICS Screen Environment screen fields.

COMMAND

For information, see Primary Commands.

DBMS

The DBMS environment. Values are:

- DL/I
- EXEC DLI
- VSAM
- SEQ
- DB2
- IDMS SQL (CA-IDMS/SQL)
- DATACOM

The value in this field is used by PREVIEW to distinguish between CALL CBLTDLI, CEETDLI, and EXEC DLI logic. It is also used to distinguish between DL/I and EXEC DLI at generation time.

TRACE

A value to specify whether the program generates and maintains trace variables for debugging. Values are:

Y

Generate trace variables.

N

(Default.) Do not generate trace variables.

Note: Trace variables increase the size of the generated program. In a production environment, the TRACE value should be N.

LINEOPT

(Not displayed for a nonterminal definition.) The line optimization logic that the program uses and generates. Values are:

1

Use the CA Telon line optimization subroutine. CA Telon performs line optimization automatically for you.

2

Simulate subroutine optimization in procedural custom code (except for fill character processing).

3

Do not generate the line optimizing code.

Note: For PWS, this value must be 2 or 3.

SPASIZE

The size of the DFHCOMMAREA used to hold the transfer work area. After CA Telon generates the program, this value appears in the CICS Program Summary section of the assembler listing as the TRANSFER AREA SIZE SPECIFIED.

For CICS screen definitions, this value must be greater than or equal to the sum of:

- The SPA header
- The application transfer work area
- The screen image size

For CICS nonterminal definitions, this value must be greater than the sum of:

- The SPA header
- The application transfer work area

Estimate a value and fine tune it as needed based on the requirements printed out on generated program listings.

For CICS nonterminal programs, you can specify **CICS ENV** to copy XFERWKA into SPA-AREA.

TRANCDE

The CICS transaction code associated with the generated program. Specify a value only when the CICS transaction is different from the CA Telon-generated program name. The CA Telon-generated program name is set as an installation default.

LNKCOPY

The name of the member containing the 01-level declarations to copy or include in the linkage section.

USGCOPY (1)

The name of the member containing the list of 01-level declarations to copy or include at the end of the calling parameter list.

USGCOPY (2) A

(Not displayed for a nonterminal definition.) The name of the member containing procedural code to copy or include at the end of the Q-100-CICS-INIT section.

BMS

(Not displayed for a nonterminal definition.) A value to specify whether to use the BMS map. Values are:

Y

Create CICS BMS source and the appropriate program code to use the BMS map

N

Use CA Telon mapping (recommended for all 3270-type terminals)

BMSMAP

(Not displayed for a nonterminal definition.) The name of the generated BMS map when it is different from the installation default name as delivered; *hhnnnn*, where:

hh

The HEADER

nnnn

The program ID

Note: The default can be customized during CA Telon installation. Verify the default with your CA Telon administrator.

If you specify a value here, you must also enter **Y** in the BMS field.

SPASTG

The section of the program where CA Telon is to generate the SPA-AREA.
Values are:

A (AUTO)

- For COBOL, CA Telon generates the SPA-AREA in the linkage section. When no SPA-AREA is specified for a program as it is executed, CA Telon initializes the SPA-AREA and the XCTL to the same program by using GETMAIN.
- For PL/I, when no SPA-AREA is specified for a program, CA Telon uses GETMAIN, sets the COMPTR, and initializes the SPA-AREA.

S (STATIC)

CA Telon generates the SPA-AREA in COBOL working storage.

IOASTG

A COBOL field that specifies where CA Telon generates the SEGMENT-IO-AREA. Values are:

A (AUTO)

Generates SEGMENT-IO-AREA in the Linkage Section and do a GETMAIN for it at program initialization.

S (STATIC)

Generates SEGMENT-IO-AREA in Working Storage.

ih1.TPBSTG field

TPBSTG

A COBOL field that specifies where CA Telon is to generate the TP-BUFFER.
Values are:

A (AUTO)

Generates the TP-BUFFER in the Linkage Section and performs a GETMAIN for it at program initialization.

S (STATIC)

Generates the TP-BUFFER in Working Storage field.

ih1.PSBSCHD field

PSBSCHD

A value to specify whether CA Telon is to automatically schedule and terminate the DL/I PSB in the CICS program sections Q-200-PSB-SCHEDULE and Q-210-PSB-TERM. Values are:

Y

(Default.) Automatically schedules and terminates the DL/I PSB.

N

Generates the Q-200 and Q-210 sections without any reference to them in the CA Telon code. In this case, you are responsible for performing the sections to schedule and terminate the DL/I PSB.

Note: The PSB name used on the schedule call is the variable name (PSB-NAME), which may be dynamically set prior to the PSB scheduling.

PSBNAME

For DL/I, the name of the PSB that the program uses.

GENPCBS

A value to specify whether CA Telon is to include DL/I PCB masks in the program. Values are:

Y

Automatically generates the PCB masks in the program.

N

You must include the PCB masks in the LNKCOPY.

D

The program is a stand-alone program that links USGCOPY TDF fields.

PLIXOPT (PL/I Only)

A field in which you can request transfer to another screen to add, update, or purge PLIXOPT statements, which override specific installation-defined PL/I defaults in the CA Telon-generated program. Values are:

C

Transfer to the Update PL/I Executable Options screen for creating a PLIXOPT statement

U

Transfer to the Update PL/I Executable Options screen for updating a PLIXOPT statement

P

Purge an existing PLIXOPT statement from this screen

Update/Show Screen Parameters

Access

On the Create/Update Screen Definition screen, enter any nonblank value in the SCRN PARMS field.

Program ID

S112

Function

Specifies these screen characteristics:

- HELP facility
- HOLD facility
- Terminal I/O characteristics
- Extended attributes characteristics

HEADER.SD UPDATE SCREEN PARMS *****

COMMAND ==> _____

GENERAL: HOLD (Y/N)

* HELP (Y/N)

UPDTA (Y/N)

APPLID _____ TELON LANGLVL

TERM IO: OUTIFIL (BLANK/UNDERSCORE/NULL)

* CAPS (ON/OFF)

* EOFKEY (Y/N)

* REFRESH (Y/N)

* ALARM (Y/N)

* EATTR (Y/N)

* EAIN _____ A EAOUT _____ B

* EALIT _____ C EAERR _____ D

Field Definitions

COMMAND

For information, see Primary Commands.

HOLD

A value to specify whether CA Telon is to generate code to build a hold facility for the system you develop. Values are:

Y

Generate the code for a hold facility

N

Do not generate the code for a hold facility

The default is the value in the HOLD field on the Update Program Definition Defaults screen.

APPLID

The application ID that the system administrator defines at the installation of CA Telon.

The default is the value in the APPLID field on the Update Program Definition Defaults screen.

LANGLVL

The version of CA Telon used to generate the program. Values are:

- 2.0
- 2.1
- 2.3
- 2.4
- 3.0
- 4.0
- 4.1
- 5.0

HELP

A value to specify whether CA Telon is to generate code to build a help facility for the system you develop. Values are:

- Y
- N

The default is the value in the HELP field on the Update Program Definition Defaults screen.

UPDTA

A value to specify whether to include in the program (by COBOL COPY or PL/I %INCLUDE) the update area regardless of data access requests. Values are:

Y

Include the update area

N

Do not include the update area

(blank)

CA Telon examines data access requests to determine whether to include the update area.

OUTIFIL

The fill character for input, outin, and select fields when they are written to the screen. Values are:

B

Fill with spaces ()

U

Fill with underscores (_)

N

Fill with low values

CAPS

A value to specify whether to translate lowercase characters input by the application user to uppercase. Values are:

ON

(*Default.*) Translate lowercase characters to uppercase.

OFF

No translation occurs on input

EOFKEY (IMS MFS Only)

A value to specify whether the application user can erase data using EOF. Values are:

Y

(Default.) Allow the user to erase data using EOF. The program recognizes through a modified data tag that data has been erased.

N

Do not allow the user to erase data using EOF. The program processes the field as if it contains data originally entered.

MFS does not return to an application program if a field has been erased using EOF when the modified data tag attribute for the field is off.

REFRESH

A value to specify whether all output fields on this screen are to be saved in the screen image portion of the transfer work area across iterations. Values are:

Y

Save output fields in the screen-image area. This value is required if the HOLD or HELP value is Y. The installation default is Y.

N

Do not save output fields in the screen-image area. This specification decreases the size of the transfer work area.

ALARM

A value to specify whether the terminal alarm is to ring automatically when an ERROR-ATTR condition is detected on output. Values are:

Y

Automatic ring on ERROR-ATTR

N

No automatic ring on ERROR-ATTR

The default is established at installation.

EATTR

A value to specify whether extended attributes are used with this screen. Extended attributes are documented in the descriptions of the EAIN, EALIT, EAOUT, and EAERR fields on this screen. Values are:

Y

Use extended attributes

N

(Default.) Do not use extended attributes.

Note: If you indicate extended attributes are used, but do not specify an attribute value in the following fields, CA Telon uses the terminal defaults specified on the Update PF Keys Definition screen. However, if you modify an attribute value already displayed on this screen to spaces by entering blanks, the attribute value is null and does not revert to the terminal default.

EAIN (color attribute)

Color attributes for input, output, and select fields. Values are:

- BLUE
- GREEN
- RED
- PINK
- TURQ (Turquoise)
- YELLOW
- NEUTRAL

See the description of the EATTR field for more information.

EAIN (highlight attribute) A

Highlight attributes for input, output, and select fields. Values are:

Valid Value	Meaning
BLINK	Field blinks when displayed.
B	
BL	

Valid Value	Meaning
REVERSE R RE REV REVER REVERS	Field displays in reverse video.
DEFAULT D DE DEF DEFAU DEFLT	Field appears in default mode.
UNDERLINE U UN UNDER	Field is underlined.

See the description of the EATTR field for more information.

EALIT (color attribute)

Color attributes for literal fields. See the description of the EALIT color attribute field for values. See the description of the EATTR field for more information.

EASLIT (highlight attribute) B

Highlight attributes for literal fields. See the description of the EALIT highlight attribute field for values. See the description of the EATTR field for more information.

EAOUT (color attribute)

Color attributes for outin and output fields. See the description of the EALIT color attribute field for values. See the description of the EATTR field for more information.

EAOUT (highlight attribute) C

Highlight attributes for outin and output fields. See the description of the EALIT highlight attribute field for values. See the description of the EATTR field for more information.

EAERR (color attribute)

Color attributes for fields flagged in error. See the description of the EALIT color attribute field for values. See the description of the EATTR field for more information.

EAERR (highlight attribute) D

Highlight attributes for fields flagged in error. See the description of the EALIT highlight attribute field for values. See the description of the EATTR field for more information.

Create/Update Driver Definitions

Access

On Online Program Definition menu, enter:

- **CR** or **UP** in the FUNCTION field
- **DR** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field

You can also access this screen from the List Data Administration Information screen by entering **U** as a line command for a listing of a driver definition.

Program ID

S210

Function

Maintains characteristics of a driver program, and accesses other TDF screens to complete the driver program definition.

```

XXXXXXXX.DR CREATE IMS/DC DRIVER DEFN * *****
COMMAND ==> _____

OPTIONS ==>  CUSTOM CODE  DATA GROUP  _  ENV IMSDRV  _
              STORED PROCEDURES  _

GENERAL: DESC _____ A REMARKS _____
* HOLD (Y/N) _____ LANGLVL _____ LANG _____ (COB/PLI)
* FRSTPGM _____ APPLID _____ UPDTA (Y/N)
* CMPLOPT _____ A IDENTIF _____ A PROCEDR _____

DATA XFERWKA _____
AREAS: A WKAREA _____

CUSTOM:
A-100 A INIT _____
C-300 A XFER _____
D-100 A TERM _____

MISC: A SECTION _____
* PGMQST _____
  
```

Fields allowing entry of multiple members

Four fields allow you to specify more than one member name:

- XFERWKA
- WKAREA
- SECTION
- PGMCUST

The XFERWKA and PGMCUST fields have no edit option field; therefore, you cannot select the custom code editor for these fields. Members entered into these fields are usually stored in shared libraries.

The WKAREA and SECTION fields have an edit option field. However, you can access the editor only when one member name exists for the field. If you have specified more than one member name for the field, go to the List/Show Custom Code screen to select the member for editing.

All four fields that accept specification of multiple members can contain a maximum of 253 bytes of data. You can specify 60 bytes for each field on this screen; you can enter **U** in the one-byte field to the right of the 60th byte to request an extension screen for additional space. When you return from the extension screen, the plus sign (+) character is displayed in the one-byte field.

Show/Purge screen

You can access the Show/Purge Driver Definitions (S214) screen from the Online Program Definition menu screen by entering:

- **SH** or **PU** in the FUNCTION field
- **DR** in the ITEM field
- *Name* in the NAME field
- *Identifier* in the ID field

Alternately, on the List Panel Definitions (P401) screen, you can enter **S** or **P** as a line command for a driver (DR).

The Show/Purge Driver Definitions screen fields are the same as the Create/Update Driver Definitions screen fields.

Field Definitions

COMMAND

For information, see Primary Commands.

OPTIONS

Other TDF functions to complete the necessary specifications of the program definition. Enter any single non-blank character in the input field to the right:

Option You Can Select	Resulting Screen Display
CUSTOM CODE	List/Show Custom Code
DATA GROUP	Create/Update Data Group
ENV <i>environment</i>	The appropriate update screen environment screen, as specified on the Update Session Controls screen
STORED PROCEDURES	List Stored Procedures to be called

DESC

The description entered on the Online Program Definition menu. You can modify the description here.

(Edit option fields) A

Fields that allow you to supply custom code member names (for example, REMARKS, WKAREA, INIT). These fields are preceded by a one-position edit option field. Enter any character in these fields to access the Custom Code Editor.

These are valid edit option values and the functions that they invoke:

U

CA Telon transfers control to a blank edit screen allowing you to create a custom code member. If you have already created a custom code member, CA Telon transfers to the List/Show Custom Code screen after you enter the custom code member name in the associated field.

If you have not specified a custom code member in the associated field, CA Telon automatically creates a custom code member and gives it the name of the corresponding entry point. For example, if you enter **U** in the edit option field for OINIT1 but have not specified a name, CA Telon names the custom code member OINIT1. The next time you access the Create/Update Screen Definition screen, the value ****DFLT**** is displayed in the name field to signify that the name matches the entry point name and that the field is protected.

O

CA Telon erases the value displayed in the associated field, including the value ****DFLT****.

This action simply eliminates the association between this entry and the custom code member. It does not delete the member. Its purpose is to allow you to rename the custom code member or associate it with another entry point.

S

CA Telon passes control to the custom code editor and returns the requested member in show mode.

REMARKS

The name of the custom code member to add to the COBOL REMARKS section of the program or to the beginning of the PL/I program.

HOLD

A value to specify whether CA Telon is to generate code to build a hold facility for the system you develop. Values are:

Y

Generate the code for a hold facility

N

Do not generate the code for a hold facility

The default is the value in the HOLD field on the Update Program Definition Defaults screen.

LANGLVL

(Protected field.) Displays the version of CA Telon that is used to generate the program. Values are:

- 2.0
- 2.1
- 2.3
- 2.4
- 3.0
- 4.0
- 4.1
- 5.0

LANG

The language of the generated programs. Values are:

COB

COBOL/LE, COBOL II

PLI

PL/I

The default is the value in the LANG field on the Update Program Definition Defaults screen.

FRSTPGM

The program ID of the first program to receive control from the driver when the transaction code is executed. The length of this field depends on naming conventions established during CA Telon installation. See your database administrator for more information.

APPLID

The application ID that the system administrator defines at the installation of CA Telon.

The default is the value in the APPLID field on the Update Program Definition Defaults screen.

UPDTA

A value to specify whether to include in the program (by COBOL COPY or PL/I %INCLUDE) the update area regardless of data access requests. Values are:

Y

Include the update area

N

Do not include the update area

(Blank)

CA Telon examines data access requests to determine whether to include the update area

CMPLOPT

Compiler parameters to be included in the generated program before the COBOL IDENTIFICATION DIVISION line or the PL/I PROC statement. The field on this screen contains 16 bytes. Once an entry has been made in the field, an extension field is presented after the field. If you need to enter a longer value, place a "U" in the extension field to go to the "Update Parameter Overflow" screen, where you can enter a total of 253 bytes, including commas.

IDENTIF

The custom code COPY member name to be added after the COBOL IDENTIFICATION DIVISION line for specification of INITIAL and other Identification Division options, or in the parentheses after OPTIONS in the PL/I PROC statement.

Note: When this copybook is used for PL/I, "MAIN" must be coded if it is desired.

PROCEDR

The custom code COPY member name to be added before the PROCEDURE DIVISION line for specification of Declaratives after the Procedure Division. This copybook is valid only for COBOL; it does not appear on the screen for a PL/I program.

XFERWKA

The name of a custom code COPY member to add to the TRANSFER WORK AREA section of the program. You can enter multiple names.

You must enter a value for this field, unless you are performing a create function and you specified a XFERWKA list on the Update Program Definition Defaults screen. In this case, that value is the default for this field.

The maximum string length for this field is 253 bytes.

WKAREA

The names of the COPY member or members that contain a definition of a work area to add to the DATA DIVISION section of the COBOL program. The COPY members can be included as part of the screen definition or be members of a library.

You can specify as many as 20 COPY members, each separated by a comma, and a maximum string length of 253 bytes.

Note: COPY members that you specify here are in addition to the standard application COPY member named *hhWKAREA*, where *hh* is the variable application header.

For information on *hhWKAREA*, see the *Programming Concepts Guide*.

INIT

The name of a COPY member containing the custom code to insert in the initialization section of the driver program, executed when the driver first receives control.

XFER

The name of a COPY member that contains the custom code to insert in the transfer control section of the driver program, executed just before control is passed to a called subroutine (for example, a screen).

TERM

The name of COPY member that contains the custom code to insert in the termination section of the driver program, executed just before the driver program returns control to the system.

SECTION

One or more names of custom code members that contain the COBOL sections or PL/I procedures that can be performed from other parts of the program. Each name must be separated by a comma. You may specify a maximum of 35 names, and a maximum string length of 253 bytes.

PGMCUST

The name of the COBOL section or PL/I procedure in which to add custom code, and the name of the custom code member added. You can make multiple specifications using this format:

section-name1, member-name1,
section-name2, member-name2,...

Section-name

The four-character identifier of the section or procedure in which to include the custom code (for example, H100) and a suffix (I or T) that specifies whether to include the code at the beginning (I) of the section or procedure, or at the end (T).

For example, H100I specifies section H100 is included at the beginning of the program and E100T specifies section E100 is included at the end of the program.

Member-name

The name of the custom code added at the location specified by *section-name*.

Thus, the value A100I,OUTIDC specifies the custom code named OUTIDC is placed at the beginning of the A-100 section.

The maximum string length is 253 bytes.

The following section names are available for Driver programs:

Section	Description
A100I	A-100-OUTPUT-INIT (beginning of section)
A100T	A-100-OUTPUT-INIT (end of section)
C100I	C-100-TERMIO-READ (beginning of section)
C100T	C-100-TERMIO-READ (end of section)

Section	Description
C200I	C-200-TERMIO-WRITE (beginning of section)
C200T	C-200-TERMIO-WRITE (end of section)
C300I	C-300-TERMIO-XFER (beginning of section)
C300T	C-300-TERMIO-XFER (end of section)
C900I	C-900-GET-SPA (beginning of section; IMS drivers with SPA)
C900T	C-900-GET-SPA (end of section; IMS drivers with SPA)
C910I	C-910-GET-MESSAGE & C-910-TERMIO-SAVE (beginning of section; IMS drivers)
C910T	C-910-GET-MESSAGE & C-910-TERMIO-SAVE (end of section; IMS drivers)
C920I	C-920-GET-WORKSPA (beginning of section; IMS drivers with WORKSPA)
C920T	C-920-GET-WORKSPA (end of section; IMS drivers with WORKSPA)
C925I	C-925-INSERT-WORKSPA (beginning of section; IMS drivers with WORKSPA)
C925T	C-925-INSERT-WORKSPA (end of section; IMS drivers with WORKSPA)
C950I	C-950-PUT-WORKSPA (beginning of section; IMS drivers with WORKSPA)
C950T	C-950-PUT-WORKSPA (end of section; IMS drivers with WORKSPA)
C960I	C-960-PUT-SPA (beginning of section; IMS drivers with SPA)
C960T	C-960-PUT-SPA (end of section; IMS drivers with WORKSPA)
C970I	C-970-PUT-MESSAGE (beginning of section)
C970T	C-970-PUT-MESSAGE (end of section)
C995I	C-995-BUFFER-INIT-LOOP (beginning of section)
C995T	C-995-BUFFER-INIT-LOOP (end of section)
D100I	D-100-INPUT-INIT (beginning of section)
D100T	D-100-INPUT-INIT (end of section)
MAINI	MAIN (beginning of section)

Section	Description
MAINT	MAIN (end of section)
MAINLINE	MAIN (replaces entire section)
MAINPROCESSI	MAIN-PROCESS (beginning of section)
MAINPROCESST	MAIN-PROCESS (end of section)
Z100I	Z-100-SECTIONS-COPY (beginning of section)
Z900I	Z-900-SECTION-FALLOUT & Z-900-PROGRAM-END (beginning of section); COBOL programs only
Z980I	Z-980-ABNORMAL-TERMINATION (beginning of section)
Z980T	Z-980-ABNORMAL-TERMINATION (end of section)

Update IMS/DC Driver Environment

Access

On the Create/Update Driver Definitions screen, enter a nonblank character to select **ENV IMSDRV** in the option field.

Program ID

S167

Function

Specifies the environment characteristics of the IMS program, the IMS PSB, the IMS MFS, the IMSDRV, and, in part, the TSO program that CA Telon generates.

```

HEADERID.DR UPDATE IMS/DC DRIVER ENV ** *****
COMMAND ==> _____

IMS:  CONVERS (Y/N)  FRSTMOD _____  PGM/PSB NAME _____
*      GENPCBS (Y/N)  LNKCOPY  (Y/N)    USGCOPY      TRACE (Y/N)

TRAN:  TRANCDE _____

SPA:   SPASIZE _____
*      WKSPASZ _____  WKSPAIO: GET _____ PUT _____  WKSPAIN (Y/N)

FLOW:  LINKPGM _____
*      (ID"S) _____
*      _____
*      MSGPGM _____
*      (ID"S) _____
*      _____
*      MSGTRAN _____
*      (ID"S, _____
*      TRAN) _____
*      MSGTBL _____  MSGBUF _____  LINKDYN (Y/N)
MSG ERR: A4EPGM _____  A4EMSG _____
STG REQ: IOASIZE _____  TPOSIZE _____  TPISIZE _____
PLIXOPT: _____ (C-CREATE/U-UPDATE/P-PURGE)

```

Field Definitions

COMMAND

For information, see Primary Commands.

CONVERS

A value to specify whether the system that CA Telon generates is IMS conversational. Values are:

Y

Generate an IMS conversational system using an IMS SPA

N

Generate an IMS non-conversational system using a WORKSPA database

FRSTMOD

The MFS MOD NAME for the screen associated with the program being generated.

PGM/PSB NAME

The name of the load module that the linkage editor creates. CA Telon requires this field only when the load module name is different from the name that CA Telon generates for the program, which is set at installation.

GENPCBS

A value to specify whether to include DL/I PCB masks in the program. Values are:

Y

Automatically generate PCB masks in the program

N

PCB masks must be included in the LNKCOPY and USGCOPY members

LNKCOPY

The name of the COPY or INCLUDE member containing the 01-level declarations to include in the linkage section.

USGCOPY

The variable declarations in the linkage member appended to the COBOL procedure division or the PL/I procedure statements.

In COBOL, the variable declaration is the list of 01-level variables in the linkage section. In PL/I, variable declaration is a list of declare statements in the linkage section.

TRACE

A value to specify whether the program generates and maintains trace variables for debugging. Values are:

Y

Generates trace variables for the CA Telon Test Facility

N

Do not generates trace variables for the CA Telon Test Facility

Trace variables increase the size of the generated program. In a production environment, the TRACE value should be N.

TRANCDE

The IMS transaction code associated with the generated program. Specify a value only when the IMS transaction is different from the CA Telon generated program name, which is set at installation.

SPASIZE

The size of the SPA specified in the IMS generation for this application. It can be defined alone or in conjunction with the WKSPASZ field value. The total of the values in the two fields must be enough bytes to hold:

- The SPA header
- The application transfer work area
- The size of the largest screen image in the application

You can fine-tune the size as needed, based on SPA requirements printed out on generated program listings.

WKSPASZ

The number of bytes in the WORKSPA database used in this application system. The value is not used when you do not use a WORKSPA database.

In a non-conversational system, the database must be large enough to hold:

- The application transfer work area
- Overhead
- The largest screen image in the application

In a conversational system, the WKSPASZ value must specify enough bytes to hold the overflow from the IMS SPA area, whose size is specified in the SPASIZE field.

WKSPAIO

The name of two copy members containing custom code used in the retrieval and writing of WORKSPA databases:

GET

Contains custom code included before the read of the WORKSPA database

PUT

Contains custom code included before the replace of the WORKSPA database

WKSPAIN

A value to specify whether the generation of CA Telon WORKSPA database initialization code is to take place in IMS program section C-920-GET-WORKSPA. Values are:

Y

Generate code to reinitialize the CA Telon transfer work area when the IMS program flow has been broken

N

Do not generate such code

LINKPGM

The program IDs of all programs to which the generated program can be dynamically linked through CA Telon, specified in this format:

id[,id.]

where *id* is the CA Telon-generated program ID in the *hhnnnn* format of your CA Telon installation.

The list of IDs may be a maximum of 256 bytes long.

Alternately, you can enter **ANY** to specify the generated program is dynamically linked by means of CA Telon to NEXT-PROGRAM-NAME if NEXT-PROGRAM-NAME-ID is not detected in a search of the MSGPGM or MSGTRAN fields. If you enter **ANY** here, do not specify ANY in the MSGPGM field or any value in the MSGTBL field.

MSGPGM

The program IDs of all programs to which this program issues IMS message switches to transfer control, specified in this format:

id[,id.]

where *id* is the CA Telon-generated program ID in the *hhnnnn* format of your CA Telon installation.

The list of IDs may be a maximum of 253 bytes long.

Alternately, you can enter **ANY** to specify that the generated program should do a message switch to NEXT-PROGRAM-NAME if the NEXT-PROGRAM-NAME-ID is not detected in LINKPGM, MSGTRAN, or MSGTBL specifications. If you enter **ANY** here, do not specify ANY for LINKPGM.

This value is valid if the transaction code for the program to receive control is the CA Telon-generated transaction code. Otherwise, specify a value in the MSGTRAN field.

MSGTRAN

The pairing of screen program IDs and corresponding IMS transaction codes, specified in this format:

id,tran[,id,tran.]

where:

id

The CA Telon-generated program ID in the *hhnnnn* format of your CA Telon installation

tran

The eight-character IMS transaction code equated with *id*

This field is limited to 253 characters. To exceed this limit, use the MSGTBL field instead.

The generated program can transfer control by issuing IMS message switches. Specify values only if the transaction code of the program to receive control is different from the CA Telon-generated transaction code.

MSGTBL

The copy member containing the list that would otherwise be entered in the MSGTRAN field.

MSGBUF

The member name and length of a user-defined buffer area for use in automatic message switching. CA Telon appends the member name to the definition for the TP-OUTPUT-XFER-BUFFER.

LINKDYN

A value to specify that CA Telon is to dynamically link to any static subroutines which are not specified in the LINKPGM, MSGPGM, MSGTBL, or MSGTRAN fields.

A value in this field is not valid if the MSGPGM value is ANY. To be dynamically called, the static subroutines must be link-edited without the NCAL option.

A4EPGM

A program ID processed as the next program when an A4 status code results from attempting an IMS message switch. The program ID must also be defined in the LINKPGM, MSGPGM, MSGTRAN, or MSGTBL field.

This field and A4EMSG are mutually exclusive.

A4EMSG

The message displayed in the ERRMSG1 field when an A4 status results from attempting an IMS message switch.

Values include literal message text or the name of the variable containing the message. Enclose literal message text in single quotes.

This field and the A4EPGM field are mutually exclusive. This field is invalid when you specify **S** in the LINKOPT field on the Update IMS/DC Driver Environment screen.

IOASIZE

The maximum size required by any one program for this segment I/O area.

TPOSIZE

The maximum size of the TP output buffer for all programs linked by means of an IMS driver.

TPISIZE

The maximum size of the TP input buffer for all programs linked by means of an IMS driver.

PLIXOPT (PLI Only)

A field in which you can request to add, update, or purge PLIXOPT statements, which override specific installation-defined PL/I defaults in the CA Telon-generated program. Values are:

C

Create a PLIXOPT statement

U

Update a PLIXOPT statement

P

Purge a PLIXOPT statement

Create/Update IMS/DC Report Definitions

Access

On the Online Program Definition menu, enter:

- **CR** or **UP** in the FUNCTION field
- **RD** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field

Program ID

S310

Function

Defines and maintains characteristics of report programs, including:

- Transfer work area name
- Other work area names
- Custom code copy names for code added to the generated program
- Report size

```

XXXXXX.RD CREATE IMS/DC REPORT DEFN * *****
COMMAND ==>
OPTIONS ==> CUSTOM CODE _ DATA GROUP _ PANEL DEF _ ENV IMS _
            STORED PROCEDURES _

GENERAL:  DESC _____ A REMARKS _____
*         L ANGLVL _____ SIZE __ X __ LANG _ (COB/PLI)
*         CMPL OPT _____ A IDENTIF _____ A PROCEDR _____
*         APPLID : _____
DATA     XFERWKA : _____
AREAS:   A WKAREA _____
*        A LINKWKA _____

OUTPUT:
A-100   A OINIT1 _____ A OINIT2 _____
B-100   A OUTTERM _____

MISC:   A SECTION _____
*        PGM CUST _____
  
```


Show/Purge screen

You can access the Show/Purge IMS/DC Report Definitions (S314) screen from the Online Program Definition menu by entering:

- **SH** or **PU** in the FUNCTION field
- **RD** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field

Alternately, on the List Panel Definitions (P401) screen, you can enter **S** or **P** as a line command for a report definition (RD).

Field Definitions

The Show/Purge IMS/DC Report Definitions screen fields are the same as the Create/Update IMS/DC Report Definitions screen fields.

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

OPTIONS

Other TDF functions to complete the necessary specifications of the program definition. Enter any single non-blank character in the input field to the right:

Option You Can Select	Resulting Screen Display
CUSTOM CODE	List/Show Custom Code
DATA GROUP	Create/Update Data Group
PANEL DEF	Update Panel Fields
ENV IMS	The appropriate update screen environment screen, as specified on the Update Session Controls screen
STORED PROCEDURES	List Stored Procedures to be called

DESC

The description of the program. The default is the value in the DESC field of the Online Program Definition menu. You can modify the description here.

(Edit option fields) A

Fields that allow you to supply custom code member names (for example, REMARKS, WKAREA, INIT). These fields are preceded by a one-position edit option field. Enter any character in these fields to access the Custom Code Editor.

These are valid edit option values and the functions that they invoke:

U

CA Telon transfers control to a blank edit screen allowing you to create a custom code member. If you have already created a custom code member, CA Telon transfers to the List/Show Custom Code screen after you enter the custom code member name in the associated field.

If you have not specified a custom code member in the associated field, CA Telon automatically creates a custom code member and gives it the name of the corresponding entry point. For example, if you enter **U** in the edit option field for OINIT1 but have not specified a name, CA Telon names the custom code member OINIT1. The next time you access the Create/Update Screen Definition screen, the value ****DFLT**** is displayed in the name field to signify that the name matches the entry point name and that the field is protected.

O

CA Telon erases the value displayed in the associated field, including the value ****DFLT****.

This action simply eliminates the association between this entry and the custom code member. It does not delete the member. Its purpose is to allow you to rename the custom code member or associate it with another entry point.

S

CA Telon passes control to the custom code editor and returns the requested member in show mode.

REMARKS

The name of the custom code member to add to the COBOL REMARKS section of the program or to the beginning of the PL/I program.

LANGLVL

The version of CA Telon used to generate the program. Values are:

- 2.0
- 2.1
- 2.3
- 2.4
- 3.0
- 4.0

SIZE

The number of lines and columns that define the size of the report. The CA Telon default is 55 lines and 120 columns per page. Once the line and the column size is established in a CA Telon session, it remains in effect for any additional reports created during that session.

LANG

The language in which CA Telon is to generate programs. Values are:

COB

COBOL/LE, COBOL II

PLI

PL/I

The default is the value in the LANG field on the Update Session Controls screen.

CMPLOPT

Compiler parameters to be included in the generated program before the COBOL IDENTIFICATION DIVISION line or the PL/I PROC statement. The field on this screen contains 16 bytes. Once an entry has been made in the field, an extension field is presented after the field. If you need to enter a longer value, place a "U" in the extension field to go to the "Update Parameter Overflow" screen, where you can enter a total of 253 bytes, including commas.

IDENTIF

The custom code COPY member name to be added after the COBOL IDENTIFICATION DIVISION line for specification of INITIAL and other Identification Division options, or in the parentheses after OPTIONS in the PL/I PROC statement.

Note: When this copybook is used for PL/I, "MAIN" must be coded if it is desired.

PROCEDR

The custom code COPY member name to be added before the PROCEDURE DIVISION line for specification of Declaratives after the Procedure Division. This copybook is valid only for COBOL; it does not appear on the screen for a PL/I program.

APPLID

The application ID that the system administrator defines at the installation of CA Telon. The default is the value in the APPLID field on the Update Program Definition Defaults screen.

XFERWKA

The custom code COPY member names to add to the TRANSFER WORK AREA section of the program.

You must enter a value for this field, unless you are performing a create function and you specified a XFERWKA list on the Update Program Definition Defaults screen. In this case, that value is the default for this field.

WKAREA

The names of the COPY member or members that contain a definition of a work area to add to the DATA DIVISION section of the COBOL program. The COPY members can be included as part of the screen definition or be members of a library.

You can specify a maximum of 20 COPY members, each separated by a comma, and a maximum string length of 253 bytes.

Note: COPY members that you specify here are in addition to the standard application COPY member named *hhWKAREA*, where *hh* is the variable application header.

For information on *hhWKAREA*, see the *Programming Concepts Guide*.

LINKWKA

The COBOL COPY or PL/I %INCLUDE member name of the link work area, which is passed from the calling program to the PRINT subroutine. The starting level for this member may not be 01.

This member allows the PRINT subroutine access to the data used in the report.

Note: If you do not pass data with this LINK-WORK-AREA, code a dummyLINKWKA. For COBOL, the recommended area name is PRINTSUB-AREA. For PL/I, the recommended field name is ADDR (PRINTSUB_AREA).

OINIT1

The name of the custom code member to place in the A-100-OUTPUT-INIT section *before* the automatic database or file read statements (that is, to perform special I/O, initialize areas, and so on).

If there is no automatic read, CA Telon still inserts this custom code member in the same section or procedure and can use the custom code member for custom database or file reads.

OINIT2

The name of the custom code member to place in the A-100-OUTPUT-INIT *after* automatic database or file read statements.

If there is no automatic read, CA Telon still inserts this custom code member in the same section or procedure and can use the custom code member for custom database or file reads.

OUTTERM

The name of the custom code member added at the end of the B-100-OUTPUT-EDITS section. This logic is performed after output edit processing and after output SEGLOOP processing (if defined).

SECTION

One or more names of custom code members that contain the COBOL sections or PL/I procedures that can be performed from other parts of the program. Each name must be separated by a comma. You may specify a maximum of 35 names, and a maximum string length of 253 bytes.

PGMCUST

The name of the COBOL section or PL/I procedure in which to add custom code, and the name of the custom code member added. You can make multiple specifications using this format:

section-name1, member-name1, section-name2, member-name2,...

Section-name

The four-character identifier of the section or procedure in which to include the custom code (for example, H100) and a suffix (I or T) that specifies whether to include the code at the beginning (I) of the section or procedure, or at the end (T).

For example, H100I specifies section H100 is included at the beginning of the program and E100T specifies section E100 is included at the end of the program.

Member-name

The name of the custom code added at the location specified by *section-name*.

Thus, the value A100I,OUTIDC specifies the custom code named OUTIDC is placed at the beginning of the A-100 section.

The maximum string length is 253 bytes.

The following section names are available for IMS/DC Report programs:

Section	Description
A100I	A-100-OUTPUT-INIT (beginning of section)
A100T	A-100-OUTPUT-INIT (end of section)
B100I	B-100-OUTPUT-EDITS (beginning of section)
B100T	B-100-OUTPUT-EDITS (end of section)
C200I	C-200-TERMIO-WRITE (beginning of section)
C200T	C-200-TERMIO-WRITE (end of section)
C940I	C-940-OUTPUT-MERGE (beginning of section)
C940T	C-940-OUTPUT-MERGE (end of section)
C970I	C-970-PUT-MESSAGE (beginning of section)
C970T	C-970-PUT-MESSAGE (end of section)
C980I	C-980-SET-DESTINATION (beginning of section)
C980T	C-980-SET-DESTINATION (end of section)
C985I	C-985-PURGE-MESSAGE (beginning of section)
C985T	C-985-PURGE-MESSAGE (end of section)
C995I	C-995-BUFFER-INIT-LOOP (beginning of section)
C995T	C-995-BUFFER-INIT-LOOP (end of section)
MAINI	MAIN (beginning of section)
MAINT	MAIN (end of section)
MAINLINE	MAIN (replaces entire section)

Section	Description
Z100I	Z-100-SECTIONS-COPY (beginning of section)
Z900I	Z-900-SECTION-FALLOUT & Z-900-PROGRAM-END (beginning of section); COBOL programs only)
Z980I	Z-980-ABNORMAL-TERMINATION (beginning of section)
Z980T	Z-980-ABNORMAL-TERMINATION (end of section)

Update IMS/DC Report Environment

Access

On the Online Program Definition menu, enter:

- **UP** in the FUNCTION field
- **EN** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field
- **IMS** in the ENVIRON field

Program ID

S168

Function

Specifies the characteristics of the IMS report program.

```
HHIIII.RD UPDATE IMS/DC REPORT ENV ** *****
COMMAND ==> 1_____

IMS:  GENPCBS  (Y/N)  LNKCOPY _____  USGCOPY _____
*     PGM/PSB  NAME  _____

MFS:  MFSMOD  _____  DEVICE

      TRACE    (Y/N)

PLIXOPT:      (C-CREATE/U-UPDATE/P-PURGE)
```

Field Definitions

COMMAND

For information, see Primary Commands.

GENPCBS

A value to specify whether to include DL/I PCB masks in the program. Values are:

Y

Automatically generate PCB masks in the program

N

PCB masks must be included in the LNKCOPY and USGCOPY members

LNKCOPY

The name of the COPY or INCLUDE member containing the 01-level declarations to include in the linkage section.

USGCOPY

The variable declarations in the linkage member appended to the COBOL procedure division or the PL/I procedure statements.

In COBOL, the variable declaration is the list of 01-level variables in the linkage section. In PL/I, variable declaration is a list of declare statements in the linkage section.

PGM/PSB NAME

The name of the load module that the linkage editor creates. CA Telon requires this field only when the load module name is different from the name that CA Telon generates for the program, which is set at installation.

MFSMOD

The MFS MOD name generated for this program if it is different from the CA Telon-generated MOD name. You can use this value to create a meaningful MOD name for users when the FORMAT command is to start the application.

DEVICE

A field in which you can request transfer to the Update IMS MFSs screen, on which you provide information for routing this report definition.

TRACE

A value to specify whether the program generates and maintains trace variables for debugging. Values are:

Y

Generate trace variables.

N

Do not generate trace variables.

Trace variables increase the size of the generated program. In a production environment, the TRACE value should be N.

PLIXOPT (PL/I Only)

A field in which you can request to add, update, or purge PLEXOPT statements, which override specific installation-defined PL/I defaults in the CA Telon-generated program. Values are:

C

Create a PLEXOPT statement

U

Update a PLEXOPT statement

P

Purge a PLEXOPT statement

```

XXXXXXXX.PD UPDATE SELECT PARMS *****
COMMAND ==>
LINE 001 COL 001 A B SIZE 024 080
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----
0001
0002
0003
-----+-----
LN COL LTH **NAME** U/S SCON SIS XFEDIT/ SEGEDIT NEXTPGM INEDIT INDBIO N
08 031 001 CHOICE01

```

LTH

(*Protected field.*) Displays the length of a select field.

NAME

(*Protected field.*) Displays the name of a select field.

U/S

The function for this select field. Values are:

U

Update the consistency edits for this select field

S

Show the consistency edits for this select field

SCONSIS

The name of the copy member containing procedural code for this select field.

XFEDIT/ SEGEDIT

A field displaying "+" save XFEDIT or SEGEDIT edits exist for this select field. If no such edits exist, this field is blank.

NEXTPGM

The program to which control is transferred if this select field is selected.

INEDIT

A value to specify whether CA Telon is to execute the E-100-INPUT-EDITS section before NEXTPGM and/or SCONSIS processing. CA Telon generates the E-100 automatically if the program contains select fields. The E-100-INPUT-EDITS section edits input fields according to FLDTYPE specification. Values are:

Y

CA Telon executes the E-100-INPUT-EDITS section automatically. If errors occur, CA Telon skips the balance of processing and returns an error message to the screen.

N

(*Default.*) CA Telon does not automatically execute the E-100-INPUT-EDITS section; however, you can explicitly call it from SCONSIS code.

INDBIO

A value to specify whether CA Telon is to perform the H-100 section to create and update segments requested for auto exec. CA Telon generates the H-100 paragraph if the program contains select fields. Values are:

Y

CA Telon automatically executes the H-100-INPUT-TERM section

N

(Default.) CA Telon does not automatically execute the H-100-INPUT-TERM section; however, you can call it explicitly from SCONSIS code.

Create/Update Nonterminal Definition

Access

On the Online Program Definition menu, enter:

- **CR** or **UP** in the FUNCTION field
- **ND** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field

Program ID

N110

Function

Maintains characteristics of a nonterminal definition and allows access to other screens to complete the definition.

```

XXXXXX.ND CREATE NONTERM. DEFINITION* *****
COMMAND=>
OPTIONS ==>CUSTOM CODE_ DATA GROUP _ PANEL DEF _ ENV CICS _
          STORED PROCEDURES _
GENERAL:  DESC _____ A REMARKS _____
          *  LONGLVL _____ SIZE __ X ____ LANG ____ (COB/PLI)
          *  CMPLOPT _____ A IDENTIF _____ A PROCEDR _____
          *  APPLID _____
FILES:    RPTDEST _____ PRNTDEST ____
DATA:     A XFERWKA _____
AREAS:    A WKAREA _____

CUSTOM:
Q-N100 A INIT1 _____ A INIT2 _____
C-N100 A GETTRAN _____
A-N100 A PRCTRAN _____
T-N100 A TERM _____

MISC:     4 SECTION 22 _____
          *  PGM CUST 23 _____
  
```

Show/Purge screen

You can access the Show/Purge Nonterminal Definition (N114) screen from the Online Program Definition menu by entering:

- **SH** or **PU** in the FUNCTION field
- **ND** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field

Alternately, on the List Panel Definitions screen, you can enter **S** or **P** as a line command for a nonterminal definition (ND).

Field Definitions

The Show/Purge Nonterminal Definition screen fields are the same as the Create/Update Nonterminal Definition screen fields.

COMMAND

For information, see Primary Commands.

You can also access the custom code editor by entering one of these commands:

CREATE *custom-code-member-name* [*member-description*]

EDIT *custom-code-member-name* [*member-description*]

UPDATE *custom-code-member-name*

SHOW *custom-code-member-name*

OPTIONS

Options for transferring to other screens to complete the nonterminal definition. Enter a nonblank character in the appropriate field:

Option You Can Select	Resulting Screen Display
CUSTOM CODE	List/Show Custom Code
DATA GROUP	Create/Update Data Group
PANEL DEF	Update Panel Fields
ENV CICS	The appropriate update screen environment screen, as specified on the Update Session Controls screen
STORED PROCEDURES	List Stored Procedures to be called

DESC

The description of the program. The default is the value in the DESC field of the Nonterminal Program Definition menu. You can modify the description here.

(Edit flags) A

The one-byte fields immediately preceding the REMARKS, WKAREA, INIT1, INIT2, GETTRAN, PRCTAN, TERM, and SECTION fields in which you can enter one of these edit requests for the corresponding copy member:

Edit Flag	Description	Comments
O	Open ¹	Opens the associated custom code entry in the nonterminal definition. Breaks the connection made to the specified custom code member(s). After an open flag, there are no COPY or INCLUDE members associated with the specified custom code entry.
S	Show	Browses the associated member(s).
U	Update ¹	Creates or updates the associated member(s). If the custom code member specified does not exist, the edit automatically processes in create mode.

Note: ¹— Not valid for PURGE and SHOW modes.

REMARKS

The name of the custom code member to add to the COBOL REMARKS section of the program or to the beginning of the PL/I program.

LANGLVL

The version of CA Telon that is used to generate the program. Values are:

- 2.0
- 2.1
- 2.3
- 2.4
- 3.0
- 4.0

SIZE

The size of (number of rows and columns on) one page of the report associated with this definition. The default is 60 X 133. This value is not used unless the definition has an associated report (that is, panel data).

LANG

The language in which CA Telon is to generate programs. Values are:

COB

COBOL/LE, COBOL II

PLI

PL/I

The default is the value in the LANG field on the Update Program Definition Defaults screen.

CMPLOPT

Compiler parameters to be included in the generated program before the COBOL IDENTIFICATION DIVISION line or the PL/I PROC statement. The field on this screen contains 16 bytes. Once an entry has been made in the field, an extension field is presented after the field. If you need to enter a longer value, place a "U" in the extension field to go to the "Update Parameter Overflow" screen, where you can enter a total of 253 bytes, including commas.

IDENTIF

The custom code COPY member name to be added after the COBOL IDENTIFICATION DIVISION line for specification of INITIAL and other Identification Division options, or in the parentheses after OPTIONS in the PL/I PROC statement.

Note: When this copybook is used for PL/I, "MAIN" must be coded if it is desired.

PROCEDR

The custom code COPY member name to be added before the PROCEDURE DIVISION line for specification of Declaratives after the Procedure Division. This copybook is valid only for COBOL; it does not appear on the screen for a PL/I program.

APPLID

The application ID that the system administrator defines at the installation of CA Telon.

The default is the value in the APPLID field on the Update Program Definition Defaults screen.

RPTDEST

The destination of the report output created by this program. Values include:

- The name of a sequential or VSAM file
- A CICS queue
- The keyword **PRINTER**

This value and any value other than PRINTER must be defined in the data group for this definition.

PRNTDEST

The four-byte name of the printer (as it is defined in the CICS TCT) to which the report from this definition is routed. A value in this field is valid only if the RPTDEST value is PRINTER.

If RPTDEST is PRINTER and you do not enter a value here, be sure to include custom code to set the variable BWA-PRINTER-ID (COBOL) or BWA_PRINTER_ID (PL/I) to the name of the printer (as it is defined in the TCT) used by the generated nonterminal program. This code should be incorporated into the initialization processing, as it is required prior to any report processing.

XFERWKA

One or more names of COPY or %INCLUDE members that contain the TRANSFER WORK AREA section of your nonterminal program. You may specify a maximum of 20 member names, each separated by a comma, and a maximum string length of 253 bytes.

This field is optional for a nonterminal program definition.

Note: Even if a value is specified in XFERWKA, no SPA-XFER-WORK-AREA is generated unless a SPASIZE is specified on the Update CICS Screen Environment screen.

WKAREA

One or more names COPY or %INCLUDE members containing the definition of work areas added to the generated program. In COBOL programs, these members are copied into the WORKING-STORAGE SECTION of the DATA DIVISION of the program.

The members can be custom code members with this definition or members of a PDS or CA-PANVALET library. You may specify a maximum of 20 member names, each separated by a comma, and a maximum string length of 253 bytes.

INIT1

The custom code placed in the beginning of the Q-N100-PROGRAM-INIT section of the generated program before the CA Telon-generated initialization code. Initialization code is generated if the RPTDEST value is PRINTER, to verify that the printer used (specified in SYSWK-PRINTER-ID (COBOL) or SYSWK_PRINTER_ID (PL/I)) is defined in the CICS TCT. See the *Programming Concepts Guide* for the nonterminal program flow chart.

INIT2

The custom code placed in the Q-N100-PROGRAM-INIT section of the generated program after generated initialization processing. See the *Programming Concepts Guide* for the nonterminal program flow chart.

GETTRAN

The custom code placed in the C-N100-GET-TRAN section of the generated program. C-N100-GET-TRAN controls TRANSACT auto exec data access; it is performed first after program initialization, and then from the main process loop after each performance of A-N100-PROCESS-TRAN. See the *Programming Concepts Guide* for the nonterminal program flow chart.

PRCTRAN

The custom code is placed in the A-N100-PROCESS-TRAN section of the generated program. A-N100-PROCESS-TRAN processes the data item (segment, record, row, or item) retrieved in C-N100-GET-TRAN. If a report is produced, A-N100-PROCESS-TRAN selects the first detail group in the nonterminal panel definition printed in the report. Selection of detail groups other than the first must be done in PRCTRAN custom code. See the *Programming Concepts Guide* for the nonterminal program flow chart.

TERM

The custom code placed in the T-N100-PROGRAM-TERM section of the generated program. This routine is performed after completion of the main process loop. See the *Programming Concepts Guide* for the nonterminal program flow chart.

SECTION

The names of custom code COPY or INCLUDE members added as COBOL sections or PL/I procedures and performed from other parts of the program.

You can specify as many as 35 members, and a maximum string length of 253 bytes.

PGMCUST

The name of the COBOL section or PL/I procedure in which to add custom code, and the name of the custom code member added. You can make multiple specifications using this format:

section-name1, member-name1,
section-name2, member-name2,...

Section-name

The four-character identifier of the section or procedure in which to include the custom code (for example, H100) and a suffix (I or T) that specifies whether to include the code at the beginning (I) of the section or procedure, or at the end (T).

For example, H100I specifies section H100 is included at the beginning of the program and E100T specifies section E100 is included at the end of the program.

Member-name

The name of the custom code added at the location specified by *section-name*.

Thus, the value A100I,OUTIDC specifies the custom code named OUTIDC is placed at the beginning of the A-100 section.

The maximum string length is 253 bytes.

The following section names are available for CICS nonterminal programs:

Section	Description
AN100I	A-N100-PROCESS-TRAN (beginning of section)
AN100T	A-N100-PROCESS-TRAN (end of section)
BN100I	B-N100-PROCESS-DETAIL (beginning of section)
BN100T	B-N100-PROCESS-DETAIL (end of section)
BN200I	B-N200-END-REPORT (beginning of section)
BN200T	B-N200-END-REPORT (end of section)
BN500I	B-N500-FORMAT- <i><groupname></i> (beginning of every B-N500-FORMAT- <i><groupname></i> section)
BN500T	B-N500-FORMAT- <i><groupname></i> (end of every B-N500-FORMAT- <i><groupname></i> section)
BN600I	B-N600-PRINT- <i><groupname></i> (beginning of every B-N600-PRINT- <i><groupname></i> section)
BN600T	B-N600-PRINT- <i><groupname></i> (end of every B-N600-PRINT- <i><groupname></i> section)

Section	Description
BN900I	B-N900-PAGE-BREAK (beginning of section)
BN900T	B-N900-PAGE-BREAK (end of section)
CN100I	C-N100-GET-TRAN (beginning of section)
CN100T	C-N100-GET-TRAN (end of section)
CN200I	C-N200-WRITE-REPORT (beginning of section)
CN200T	C-N200-WRITE-REPORT (end of section)
MAIN_PROCESS_LOOP	MAIN_PROCESS_LOOP (beginning and end of procedure; PL/I CICS Nonterminal)
MAINI	MAIN (beginning of section)
MAINT	MAIN (end of section)
MAINPROCESSI	MAIN-PROCESS (beginning of section)
MAINPROCESST	MAIN-PROCESS (end of section)
QN100I	Q-N100-PROGRAM-INIT (beginning of section)
QN100T	Q-N100-PROGRAM-INIT (end of section)
QN200I	Q-N200-PSB-SCHEDULE (beginning of section)
QN200T	Q-N200-PSB-SCHEDULE (end of section)
QN210I	Q-N210-PSB-TERM (beginning of section)
QN210T	Q-N210-PSB-TERM (end of section)
QN300I	Q-N300-ACQUIRE-WORKAREAS (beginning of section)
QN300T	Q-N300-ACQUIRE-WORKAREAS (end of section)
TN100I	T-N100-PROGRAM-TERM (beginning of section)
TN100T	T-N100-PROGRAM-TERM (end of section)
Z100I	Z-100-SECTIONS-COPY (beginning of section)
Z900I	Z-900-SECTION-FALLOUT & Z-900-PROGRAM-END (beginning of section); COBOL programs only)
Z970I	Z-970-IDMSSQL-ERROR (beginning of section)
Z970T	Z-970-IDMSSQL-ERROR (end of section)
Z980I	Z-980-ABNORMAL-TERMINATION (beginning of section)

Section	Description
Z980T	Z-980-ABNORMAL-TERMINATION (end of section)
Z990I	Z-990-PROGRAM-ERROR (beginning of section)
Z990T	Z-990-PROGRAM-ERROR (end of section)

List Environments

Access

On the Online Program Definition menu, enter:

- **LI** in the FUNCTION field
- **EN** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field

Program ID

S401

Function

Displays a list of existing environments for an SD or DR program.

```
LIST HNNNN.SD ENVIRONMENTS *****
COMMAND ==>
***ENV*** **STATUS** ***** USER  UPDATE
- CICS_____ *ENV SAVED      xyzab020 041605
- IMS_____      xyzab020 041605
```

Field Definitions

COMMAND

For more information, see the chapter "User Profile Maintenance."

FUNCTION

A column position to the left of the NAME field in which you can enter a control character to manipulate the item on the line. Values are:

- **U** - Update
- **S** - Show
- **P** - Purge with confirmation
- **Z** - Zap (delete without confirmation)

ENV

The environment type.

STATUS

Displays the status of the environment after action is taken, depending on the value in the Function field.

Values, their meanings, and the associated messages are:

Value	Description	Message
U	Update	*ENV SAVED
S	Show	*ENV SHOWN
P	Purge	N/A (no longer displayed)
Z	Zap	*PURGED

USER

(*Protected field.*) Identifies the last user to access the member.

UPDATE

(*Protected field.*) Identifies the date of the last access of the member.

Select the Stored Procedures option on any of the following screens:

Program ID

S225

Function

Allows specification of one or more stored procedures to be called by the current program being defined.

[illegible]

Field Definitions

COMMAND

For more information, see *Primary Commands* in Editors and Commands.

FUNCTION

One of the following may be specified on any line containing a stored procedure entry:

Code	Function	Description
U	Update	Branch to the Create/Update Stored Procedure Program Definition screen B210.
S	Show	Branch to the Show/Purge Stored Procedure Program Definition screen B214.
Z	Zap	Delete the stored procedure call from the current program being defined.

ENTITY (output-only)

The entity name for the stored procedure to be called by the current program being defined.

NAME (output-only)

The external name (if specified) for the stored procedure to be called by the current program being defined.

RESULTS (output-only)

The number of result sets (if any) being returned from the specified stored procedure.

PRESP

The name of a custom code member to place in the S-100-CALL-*spname* paragraph prior to the call to the stored procedure with a name of *spname*.

POSTSP

The name of a custom code member to place in the S-100-CALL-*spname* paragraph after the call to the stored procedure with a name of *spname*.

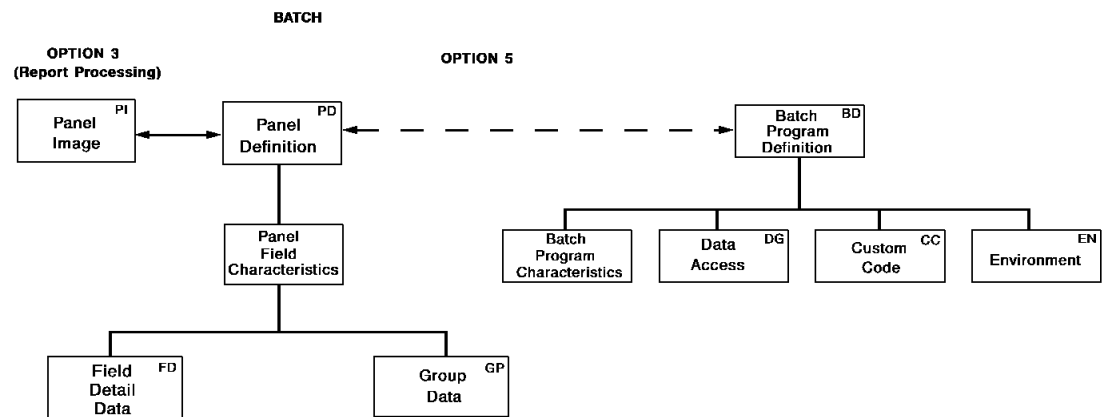
IGNORE

A comma-separated list of IGNORE codes to be included in the check of the SQLCODE return value after the call to the stored procedure *spname*, in addition to the check for a value of +466 (indicating the presence of at least one result set). A single IGNORE value of `_ALL_` may be specified, that essentially leaves the IGNORE checking up to the user; otherwise, ignore codes must be numeric, with a leading `_+_` or `_ -_` sign.

Chapter 7: Batch Program Definition

This chapter documents the screens used to create a batch program or stored procedure definition, Option 5 on the TDF Main Menu. It also discusses other screens that you use to create a panel definition and program definition for a batch program.

The following diagram shows the components of a batch program definition:



If the batch program produces a report, the first steps in creating the program are to create a panel image and a panel definition. If not, you begin with the Batch Program Definition menu.

Note: Previously-exported batch program definitions can be imported into the TDF for maintenance using CA Telon definition import utilities defined in Programming Concepts Guide.

Update Panel Fields (Batch)

Access

On the Panel Definition menu, enter:

- **CR** or **UP** in the FUNCTION field
- **PD** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field

If you did not specify a group when you created the panel image, CA Telon displays the Panel Definition menu. From there, access the Line Edit screen and specify a group CA Telon displays the Update Panel Fields (Batch) screen.

You can also access this screen from the Create/Update Nonterminal Definition screen by entering a nonblank character to select PANEL DEF.

Program ID

P255

Function

Specifies the characteristics of all the fields in the panel image, the first step in creating a panel definition.

```

TRBD11.PD UPDATE PANEL FIELDS *****
COMMAND ==>                                     PAGE 01 MORE
LINE 001 COL 002 *****                          SIZE 055 133
LINE  ---1---2---3---4---5---6---7---
0001 *** GROUP *****
0002   TRBD11                      EMPLOYEE LIST BY ID >>>
0003

U LN COL LTH USE **NAME** *FLDTYPE* ***** DBNAME OR TEXT ***** REQ MORE
GP A          TOPPAGE
i 01 053 008 OU
  01 070 002 OU
GP B          DETAIL
  01 004 002 OU
  01 008 006 OU
  01 016 020 OU
  01 038 012 OU
  01 052 003 OU
  01 058 007 OU
  01 067 005 OU
  02 016 025 OU
  03 016 025 OU
  03 044 002 OU
  03 049 005 OU

```

Field Definitions

COMMAND

For information, see Primary Commands and Update Panel Fields (Online).

LINE COL

(Protected field.) Identifies the starting line position and starting column position of the field. In a batch panel definition, these values represent the location of the field in its group, not in the report.

SIZE

The size of the field.

U

A field in which you can specify update or deletion of any field in this panel definition. Values for a batch definition are:

D

Delete this item

U

Display the appropriate screen for this item:

- Update Batch Output Fields
- Update Batch Literal Fields

I

Insert a blank line to add a field or literal to the image

LN COL LTH

The starting line position, starting column position, and length of the field. In a batch panel definition, these values represent the location of the field in its group, not in the report.

Note: When the USE value is GP, these fields are blank and protected.

USE

The use of the field. Values are:

LI

Literal field

OU

Output field

GP

This field identifies the beginning of a batch group

NAME

The name of the field.

GPTYPE OR FLDTYPE

For a literal or output field, the type of editing performed on this field. See Update Panel Fields (Online) for values.

For a separator, the type of batch group. Values are listed in the following table.

Batch Group	Description
TOPPAGE	Page heading. This group is printed only at the top of a report page.
TOPDTL	Detail heading. This group is printed only at the top of a detail group.
DETAIL	This group is an output detail.
BOTPAGE	Page footing. This group is printed only at the bottom of a report page.
CONTROL	This group defines a control break.
SUMMARY	Report footing. This group is printed at the end of the report. Only one summary group is allowed in a batch definition.

In the following example:

- TITLE is a page heading (TOPPAGE).
- EMPDET is a detail group. It defines a detail line of the report.
- MORE in the upper-right corner indicates that there is at least one more page of data for this field listing. Page down to view the next screen of data.

```

TRBD11.PD UPDATE BATCH PANEL FIELDS * *****
COMMAND ==>                                     PAGE 01 MORE
LINE 001 COL 002 ***** SIZE 055 133
LINE 1-----2-----3-----4-----5-----6-----7-----
0001 *** GROUP NAME=TITLE TYPE=TOPPAGE *****
0002 TRBD11 EMPLOYEE LIST BY ID >>>>>>> PAGE >>
0003

      GPTYPE OR
U LN COL LTH USE **NAME** *FLDTYPE* ***** DBNAME OR TEXT ***** REQ MORE
      GP TITLE TOPPAGE
01 053 008 OU CURDATE CURRENT-DATE
01 070 002 OU PAGENO NONE
      GP EMPDET DETAIL
01 004 002 OU SEQNUM
01 008 006 OU ID EMPL-ID
01 016 020 OU NAME EMPL-NAME
01 038 012 OU PHONE EMPL-PHONE
01 052 003 OU DEPT EMPL-DEPARTMENT
01 058 007 OU RATE NUMERIC EMPL-HOURLY-RATE
01 067 005 OU HRWEEK NUMERIC EMPL-HOURS-PER-WEEK
02 016 025 OU STREET EMPL-STREET
03 016 025 OU CITY EMPL-CITY
03 044 002 OU STATE EMPL-STATE
03 049 005 OU ZIP EMPL-ZIP

```

DBNAME OR TEXT

The mapping name. On an output field, it indicates the program data field that is mapped out.

REQ

(Not applicable to a batch panel field definition.)

MORE

A field that allows CA Telon to signify, by displaying a plus sign (+), that there is additional detail data for this field that cannot be displayed on this screen.

To display the additional data (or enter additional data), enter **U** in the U field; this transfers you to the appropriate screen:

- Update Batch Literal Fields
- Update Batch Output Fields

Update Batch Literal Fields

Access

On the Update Panel Fields (Batch) screen, enter **U** in the U field of a listing for a literal field. Alternatively, on the Panel Definition menu, enter **UP** in the FUNCTION field and specify values in other fields on the menu that identify the literal field to be updated.

After you press End to save your entries, this screen displays data on the next literal field that you selected for update (on the Update Panel Fields (Batch) screen). If there are no more literal fields to update, control returns to the screen from which you accessed this screen.

Program ID

P280

Function

Updates the definition of a literal field in a batch program report.

```
XXXXXX.PD UPDATE LITERAL FIELD *****
COMMAND ==> _____

FIELD NAME _____ USAGE LITERAL LINE 403 COL 026 LTH 020

MAPPING: TEXT SAMPLE BATCH PROGRAM_____
*
*
*
*      MAPOUT BLANK.WHEN.SAME _ (Y/N)
```

Field Definitions

COMMAND

For information, see Primary Commands.

FIELD NAME

The name of the field, originally defined on the Edit Panel Image screen or the Line Edit screen. You can rename the field here.

USAGE

The usage of the field. The value (always LITERAL) is established in the USE field on the Update Panel Fields (Batch) screen.

LINE COL LTH

(Protected field.) Identifies the starting line position, starting column position, and length of the field. In a batch panel definition, these values represent the location of the field in its group, not in the report.

TEXT

The text of the literal field.

BLANK.WHEN.SAME

A field that allows you to specify spaces on output when the value in the field has not changed between the last printing of the line and the current printing of the line.

Y

Automatically sets the field to spaces on output when it has the same value as the previous detail line

N

Do not automatically set the field to spaces

Update Batch Output Fields

Access

On the Update Panel Fields (Batch) screen, enter **U** in the U field of a listing for an output field. Alternatively, on the Panel Definition menu, enter **UP** in the FUNCTION field and specify values in other fields on the menu that identify the literal field to be updated.

After you press End to save your entries, this screen displays data on the next output field that you selected for update on the Update Panel Fields (Batch) screen. If there are no more output fields to update, control returns to the screen from which you accessed this screen.

Program ID

P281

Function

Updates the definition of an output field in a batch program report.

```

XXXXXX.PD UPDATE OUTPUT FIELD *****
COMMAND ==> _____

FIELD NAME TESTDIL  USAGE  OUTPUT  LINE  01 COL 002 LTH 008

MAPPING: DBNAME _____
*          OF _____
*          INIT _____
*          MAPOUT _____
*          BLANK.WHEN.SAME (Y/N)
GENERAL: PIC _____

EDIT: FLDTYPE _____ PARM LIST EXTENSION __
*          SPEC _____ (FORMAT/CONVERT)
*          _____
*          _____
*          _____
*          _____

ALT  CNTGRP _____ SCOPE _____ GROUP _____
MAPPING: TOTREF _____ SCOPE _____ GROUP _____
*          CALC _____
*          _____
*          _____
*          _____

```


Field Definitions

COMMAND

For information, see Primary Commands.

FIELD NAME

The name of the field, originally defined on the Edit Panel Image screen or the Line Edit screen. You can rename the field here.

USAGE

The usage of the field. The value is established in the USE field on the Update Panel Fields (Batch) screen. If you specify **LITERAL** and press Enter, control transfers to the Update Batch Literal Fields screen.

LINE COL LTH

(Protected field.) Identifies the starting line position, starting column position, and length of the field. In a batch panel definition, These values represent the location of the field in its group, not in the report.

DBNAME

The name of the field (in the file description or work area) from which the data is mapped.

Note: To have CA Telon generate special code to support the COBOL II "ACCEPT ... FROM DATE" or COBOL for MVS and VM "MOVE FUNCTION CURRENT-DATE ..." use the reserved word **@DATE**. This reserved word is appropriate for OUTPUT only; it should not be used for INPUT-only fields. A separate INPUT DBNAME must be specified for OUTIN fields.

OF

(COBOL only.) The name of high-level qualifiers for work areas containing the data field names defined by the specification in the DBNAME field.

INIT

The initialized value of the output field when there is no data written to it. If a value is specified here, the BLANK.WHEN.SAME field value must be N.

MAPOUT

The name of a COBOL or PL/I field in the CA Telon program whose value determines whether the field is mapped to the output buffer. When the value in the program field is Y, CA Telon maps the field to the output buffer. Otherwise, CA Telon does not map the field.

BLANK.WHEN.SAME

A field that allows you to specify spaces on output when the value in the field has not changed between the last printing of the line and the current printing of the line.

Y

Automatically sets the field to spaces on output when it has the same value as the previous detail line

N

Do not automatically set the field to spaces

PIC

The COBOL or PL/I picture clause that identifies display control on output for NUMERIC, FULLNUM, or DOLLAR fields. You can specify any valid COBOL or PL/I numeric format control characters.

Note: If the value in the FLDTYPE field is FLOAT, the value in the PICfield is ignored.

FLDTYPE

The field editing performed on the field. When you specify a DBNAME value, the default is ALPHA; otherwise, the default is NONE.

Note: Use of this field is not valid when CONVERT or FORMAT is used in theSPEC field.

PARM LIST EXTENSION

A field in which you can request transfer to the Update Parameter List Extension screen. Use this screen to specify extended fields for installation-defined FLDTYPE edits. A value must exist in the FLDTYPE field.

To transfer, enter any nonblank character.

SPEC

The edit specification:

FORMAT

Defines a mask used to format an alphanumeric field on input or output.

CONVERT

Defines acceptable display values and specifies how the program stores them. It can be used for input or output mapping.

Enter the specification in the field and its associated values in the extended field space below the SPEC field. The syntax for entering each specification and a full description of its function follow.

FORMAT *mask*

In *mask*, 9s represent numbers and Xs represent characters. All other characters are inserted into the corresponding positions on output and stripped from the corresponding positions on input. Here are examples of how an input value is stored after formatting by the mask and how that same value is displayed on output through the same mask:

Input	Mask	Stored value	Output
123-45-678	999-99-999	12345678	123-45-678
ABCD343	ABXX993	CD34	ABCD343
RTXY887	ABXX993	XY88	ABXY883
CDQ34	XXT99	CD34	CDT34

All characters except the mask-defined value 9 are stripped for storage. CA Telon does not verify the value in X position on input. (In the third example, CA Telon accepts but does not store the input value RT.) CA Telon does verify that characters specified as 9s are numbers.

The length of *mask* must equal the length of the input field. If the data field is longer, it is blank-padded to the right during mapping. Remember, the target data field must be alphanumeric even though the mask is all 9s.

CONVERT *screen-val-1,stored-val-1* [, *screen-val-2,stored-val-2* ...]

Screen-val is the value as it appears on the screen; *stored-val* is the value as it is stored. If you define more than one pair, the length of all occurrences of *screen-val* must be the same; this is also true for all occurrences of *stored-val*. If you use blanks to pad the length, enclose the value in single quotes.

In the following example, the two pairs specified result in the indicated screen value and stored value:

```

CONVERT FEMALE,F,'MALE ',M
Valid screen value    Stored value
FEMALE               F
MALE                 M

```

For input or select fields, CA Telon puts the pairs in a table to be searched at runtime. During execution, the application user's input must match a specified *screen-val*. If not, CA Telon flags the field and an error is returned.

If there is no corresponding *stored-val* during output, the program displays the stored value as is.

RANGE *start-range-1,end-range-1 [,start-range-2,end-range-2 ...]*

Start-range and *end-range* can be numeric constants or data field names. Values must be specified lowest to highest. The range is inclusive; that is, the numbers defining the range are themselves within the range.

In the following example, valid input values for the field are defined as between 4 and 9, 23 and 33, or 53.8 and 75.

RANGE 4, 9, 23, 33, 53.8, 75

CNTGRP

The name of a group whose detail occurrences are counted, with the result displayed in the field identified in FIELD NAME. Alternatively, you can specify **ALLDETAIL** for all groups to be counted.

The counter is incremented by 1 each time the group specified in CNTGRP field is printed. It is initialized when the count itself is printed (that is, when the field specified in FIELD NAME is printed), unless you specify otherwise in the SCOPE field.

If you specify a group here, do not specify a value for the DBNAME, FLDTYPE, SPEC (CONVERT), CALC, or TOTREF fields.

SCOPE

The indicator of the point at which the count (if CNTGRP contains a value) or the total (if TOTREF contains a value) is initialized. Values are:

GROUP

Initialize the count or total each time the group specified in the GROUP field on this screen is printed

PAGE

Initialize the count or total at the end of each page

REPORT

Initialize the count or total only at the beginning of the report

GROUP

The name of a group, when the value in the SCOPE field is GROUP. (A value here is ignored if the value in the SCOPE field is not GROUP.)

TOTREF

The name of a field name whose values are totaled with the result displayed in the field identified in FIELD NAME. This must be the name of a field with a numeric DBNAME or a CALC specification; it must belong to this batch definition.

The total is incremented by the value in the field identified in TOTREF each time the field is printed. It is initialized when the total itself is printed (that is, when the field specified in FIELD NAME is printed), unless you specify otherwise in the SCOPE field.

If you specify a field here, do not specify a value for the DBNAME, FLDTYPE, SPEC (CONVERT), CNTGRP, or CALC fields.

TOTSIZE

Two values representing the size of the variable used to map to the TOTREF fieldname.

The first value is the number of positions to the left of the decimal point; the second value is the number of positions to the right of the decimal point. The default size is 9(11)V9(7) for COBOL and 10(9)V4(9)T for PL/I.

A value in this field is not valid without a value in the TOTREF field.

CALC

The COBOL or PL/I expression to calculate the value contained in the field specified in FIELD NAME. The expression may refer to program-defined variables only and not to CA Telon field names specified on the Update Panel Fields (Batch) screen.

If you specify an expression here, do not specify a value for the DBNAME, FLDTYPE, SPEC (CONVERT), CNTGRP, or TOTREF fields.

Update Panel Group

Access

Access this screen in one of these ways:

- On the Update Panel Fields (Batch) screen, enter **U** in the U field of a listing for a group field (that is, the USE field value is GP)
- On the Line Edit screen, enter **U** in the line number field displaying the =GRP=> message
- On the Batch Program Definition menu, enter **UP** in the FUNCTION field, **GP** in the ITEM field, and name the group

After you press End to save your entries, this screen displays data on the next group that you selected for update (on the Update Panel Fields (Batch) screen). If there are no more groups to update, control returns to the screen from which you accessed this screen.

Program ID

P290

Function

Specifies additional characteristics of fields with a usage of GP.

```

XXXXXX.PD UPDATE PANEL GROUP *****
COMMAND ==> _____

GROUP NAME GRP1__ TYPE TOPPAGE

GENERAL: SKIPBEF ____ (NN/PAGE)  SKIPAFT ____ (NN/PAGE)
*      FMTCUST _____
*      PRINT  _____

DETAIL: TDSKIP  _
*      REPSEQ  - - - - -

CONTROL: CTLVAR _____
*      CTLLTH ____ CTLPIC _____
*      MINOR  _____

GRP REF: (TOPPAGE, TOPDTL, CONTROL)
*      FORGRP ALLDETAIL _____
*      _____
*      _____
*      _____

```

For information, see Primary Commands.

Field Definitions

GROUP NAME

The one- to eight-character name of the group. It is established on the Update Panel Fields (Batch) screen but can be modified here.

The value in this field must begin with an alphabetic character and contain no special characters. It is used to generate section name
B-5000-FORMAT-*groupname* and B-6000-PRINT-*groupname*.

TYPE

The type of batch group. Values are:

TOPPAGE (Page heading)

This group is printed only at the top of a report page.

TOPDTL (Detail heading)

This group is printed only at the top of a detail group.

DETAIL

This group is an output detail.

BOTPAGE (Page footing)

This group is printed only at the bottom of a report page.

CONTROL

This group defines a control break.

SUMMARY (Report footing)

This group is printed at the end of the report. Only one summary group is allowed in a batch definition.

SKIPBEF

The number of lines to skip *before* printing this group. Values are:

nn

Skip *nn* lines before printing

PAGE

Go to the next page before printing

SKIPAFT

The number of lines to skip *after* printing this group. Values are:

nn

Skip *nn* lines after printing

PAGE

Go to the next page after printing

FMTCUST

Custom code inserted at the end of section B-5000-FORMAT-groupname

PRINT

A value to specify whether the group is printed. Enter the name of the variable which controls printing of the group.

You must control the value in this field using custom code. The value of the variable name can be set in custom code to Y (print the group) or N (do not print the group).

TDSKIP

The number of lines to skip before printing a detail group when the previously printed group is not the current group (that is, when a page break, control break, or detail change occurs). A value in this field is meaningful only when the TYPE value is DETAIL.

REPSEQ

The repeating sequence of line skips when the previously printed group is the same as this group.

For example, if you enter 1 2 2 1, the repeating sequence is:

Lines	After
skipped	printing of:
1	First group
2	Second group
2	Third group
1	Fourth group
1	Fifth group
2	Sixth group
2	Seventh group
1	Eighth group
. .	
. .	
. .	

CTLVAR

The name of the control variable name that the program must check for control breaks. When the variable's value changes, the group is printed.

This field is required when the TYPE value is CONTROL, but is otherwise not meaningful.

CTLLTH

The length of the control variable checked for a control break. You can use this field to cause a control break on a field shorter than what is specified in CTLVAR.

This field is required when the TYPE value is CONTROL, but is otherwise not meaningful.

CTLPIC

The picture clause used to check the control variable named in the CTLVAR field. If you specify a picture clause, the value in CTLLTH must specify the actual length in the core of the CTLPIC clause.

This field is valid only when the value in the TYPE field is CONTROL.

MINOR

A minor control group; that is, a control group printed in the report before printing the control group whose definition is currently displayed.

This field is valid only when the value in the TYPE field is CONTROL.

FORGRP

A detail group or a list of detail groups that are logically related to the control group whose definition is currently displayed. The reserved word "ALLDETAIL" (the default) can be used to indicate all detail groups are logically related to the control group.

The type of the group specified here must be DETAIL. The type of the group currently displayed must be TOPPAGE, TOPDTL, or CONTROL.

This value is applied differently depending on the type of the currently displayed group:

- For a TOPPAGE group, this value specifies detail groups that require the TOPPAGE heading when printed as the first detail group on the page. Thus, you can specify this value to create continuation headings or to suppress a heading on a summary page.
- For a TOPDTL group, this value specifies detail groups that require the TOPDTL heading when the previous group printed in the report is unlike the detail group(s) specified (that is, the TOPDTL heading is to be printed before the detail group when a page break, control break, or detail change occurs).
- For a CONTROL group, this value specifies the detail groups which, when printed, require that this control break be checked (that is, for which details is the control break valid). If you omit a value, control break checking takes place regardless of the detail group being printed.

Batch Program Definition

Access

On the TDF Main menu, enter **5** in the FUNCTION field. You can return to the Update Session Controls menu by pressing PF3.

Program ID

B100

Function

Specifies values required for a batch or stored procedure program definition.

Short form

If the value in the USER MODE field of the Update Session Controls screen is 1 or spaces, the short form of the Batch Program Definition menu is displayed:

BATCH PROGRAM DEFINITION MENU *****					
COMMAND ==> _____					
FUNCTION:	_	CR-CREATE	UP-UPDATE	PU-PURGE	SH-SHOW
ITEM:	_	BD-BATCH	SP-STORED PROCEDURE		
MEMBER NAME:					
	HEADER	_____			
	ID	_____			
	DESC	_____			
BASE DEFN : _____ (FOR CREATE - NAME OF BASE BD OR SP)					

Switching menu mode

Depending on the value in the USER MODE field of the Update Session Controls screen, CA Telon displays the long form or the short form of this menu. See Update Session Controls for more information.

To switch from one to the other, enter **SETMODE** in the COMMAND field or change the USER MODE value on the Update Session Controls screen in the User Profile Maintenance chapter.

Long form

If the value in the USER MODE field of the Update Session Controls screen is 2, the long form of the Batch Program Definition menu is displayed:

```

BATCH PROGRAM DEFINITION MENU *****
COMMAND ==> _____

FUNCTION: _  CR-CREATE  UP-UPDATE  PU-PURGE  SH-SHOW  LI-LIST

ITEM:      -  BD-BATCH  SP-STORED PROCEDURE  PI-IMAGE  PD-DEFIN
              DG-DATA GROUP  CC-CUSTCODE  EN-ENVIRON  GP-GROUP

MEMBER NAME:
  HEADER  _____
  ID      _____  TYPE __ (BD, SP)
  DESC    _____

BASE DEFN : _____ (FOR CREATE - NAME OF BASE BD OR SP)

ENTER VALUE FOR SPECIFIC ITEM TO BE PROCESSED:
  1. GROUP      _____ (NAME OF GROUP)
  2. CUSTCODE   _____ (NAME OF CUSTOM CODE)
  3. ENVIRON    _____ (MVS, STORED)

```

Field Definitions

COMMAND

For information, see Primary Commands.

You can also enter the SETMODE command to swap to the long form of the menu.

FUNCTION

The type of function to perform on the definition. This table shows valid values with respect to the type of item being defined.

Function (Action)	BD	SP	PI	PD	DG	CC	EN	GP
CR (Create)	Y	Y	N	N	Y	Y	Y	N
UP (Update)	Y	Y	Y	Y	Y	Y	Y	Y
PU (Purge)	Y	Y	Y	Y	N	Y	N	N
SH (Show)	Y	Y	Y	Y	N	Y	N	N
LI (List)	Y	Y	Y	Y	Y	Y	N	N

ITEM

The item that is the subject of the function specified in the FUNCTION field. Values are:

BD

Batch definition

SP

Stored Procedure definition

PI

Panel image

PD

Panel definition

CC

COBOL or PL/I custom code

EN

Program environment

GP

Group

DG

A group of databases, data sets, or IMS TP PCBs

Note: On the short form of the Batch Program Definition menu, only BD is a valid value.

HEADER

The character string that identifies the prefix to be processed, the program ID, or group of programs. Its length is determined at installation.

You can list the members associated with this header. For example, you can request a list of members associated with the header ZZ by specifying:

- LI in the FUNCTION field
- BD in the ITEM field
- ZZ in the HEADER field

ID

The one- to five-character name which uniquely identifies the definition within a group sharing the same header. The length is set at installation.

TYPE

The type of program definition. Values are:

BD

Batch definition

SP

Stored Procedure definition

DESC

The description. If you accessed this screen from the Panel Definition menu, the description is carried over from that menu; otherwise, this field is blank. A description is optional.

BASE DEFN

An existing batch definition used as a base for the definition you are creating. Enter the concatenation of the HEADER and ID values that identify the base definition to be copied.

The copy action is initiated only if the FUNCTION field value is CR.

GROUP

The name of a group created with the panel definition function. A value is required when you request updating a specific group of the panel definition by specifying **UP** in the FUNCTION field and **GP** in the ITEM field.

CUSTCODE

The name of the custom code member you are creating or updating. A value is required; in the FUNCTION field, specify either:

- CR
- UP
- PU

and CC in the ITEM field.

ENVIRON

The target environment for the batch program. Value is:

- MVS (MVS, z/OS)

Create/Update Batch Definitions

Access

On the Batch Program Definition menu, enter:

- **CR** or **UP** in the FUNCTION field
- **BD** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field

Program ID

B110

Function

Specifies values required to create a batch program, including:

- The name of the program work area custom code
- The printed output size
- The names of other custom code members added to the program

To save your entries and return to the Batch Program Definition menu, press End.

```

TRC211.BD UPDATE BATCH DEFINITION *** *****
COMMAND ==>
OPTIONS ==> CUSTOM CODE _ DATA GROUP _ PANEL DEF _ ENV MVS _
            STORED PROCEDURES _
GENERAL: DESC _____ A REMARKS _____
*   LANGLVL 5.0_          SIZE _ X _          LANG _ (COB/PLI)
*   STRUCTURE:  + STANDARD _ MAIN SORT _      MERGE _ MATCH
*   APPLID _____ USER SORTS
*   CMPLOPT _____ A IDENTIF _____ A PROCEDR _____
FILES:  RPTDEST _____
*   A COBFCPY:SELECT _____ FILEDEF _____

AREAS:  A WKAREA _____

Q-1000  A INIT1 _____ A INIT2 _____
C-1000  A GETTRAN _____

A-1000  A PRCTRAN _____

T-1000  A TERM _____

MISC:   A SECTION _____
*       PGMCUST _____
LINKAGE: PARMs _____

```

Note: If you press Enter to save your entries, this screen is redisplayed. However, depending on the structure you have defined, certain custom code points may not be redisplayed.

Show/Purge screen

You can access the Show/Purge Batch Definitions (B114) screen from the Online Program Definition menu by entering:

- **SH** or **PU** in the FUNCTION field
- **BD** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field

Alternatively, on the List Panel Definitions screen, you can enter **S** or **P** as a line command for a batch definition (BD).

The Show/Purge Batch Definitions screen fields are the same as the Create/Update Batch Definitions screen fields.

Field Definitions

COMMAND

For information, see Primary Commands.

You can also enter one of the following commands to invoke the custom code editor for a specified member:

- **CREATE** *member-name* [*member-description*]
- **EDIT** *member-name* [*member-description*]
- **UPDATE** *member-name*
- **SHOW** *member-name*

Note: *Member-description* is optional.

OPTIONS

The screen to which you transfer to complete the necessary specifications of the program definition.

The next table shows the options by field name and the associated screen to which control transfers. Select a field by entering a nonblank character.

OPTIONS Field	Screen to Which Control Transfers
CUSTOM CODE	List/Show Custom Code
DATA GROUP	Update Database Segment
PANEL DEF	Update Panel Fields (Online)
ENV	Update Batch Environment
STORED PROCEDURE	List stored procedures to be called

DESC

The description entered on the Batch Program Definition menu. You can change the description here.

(EDIT Flag fields) A

A one-byte field immediately succeeding several fields, as shown on the screen illustration. Values are presented following:

Value	Meaning	Remarks
O	Open ¹	Allows you to rename the custom code member or associate it with another entry point. It erases any value displayed in the field's name field (this is also true for the name "***DFLT***"). This action only disconnects the association between this entry and the custom code member. It does not delete the member's contents.
S	Show	Allows you to browse the associated member name.
U	Update ¹	Allows you to update or create the member name.

Note: ¹— Not valid for purge or show modes. Open some members and show others.

REMARKS

(Protected field.) Identifies the name of the custom code member added in the COBOL REMARKS section of the program or at the beginning of the PL/I program.

LANGVL

The CA Telon release version. For batch, the default is the current release at your site.

SIZE

The length and width of the report. The default is 60 lines x 133 columns.

Note: The size of the panel definition has no bearing on the size of the report.

LANG

The programming language in which the definition is generated. This field is used only when both COBOL and PL/I options are installed with CA Telon. The value specified here overrides the default-entered LANG field on the Update Program Definition Defaults screen.

STANDARD

A field in which a plus sign (+) indicates that the original batch program structure is the current structure of the program.

If the structure is standard, you can modify it using the MAINSORT, MERGE, or MATCH field.

MAIN SORT

A field in which a plus sign (+) indicates that mainline sort is the current structure of the program. You can overwrite + with one of these values:

U

Transfer to the Update Sort Definition screen to enter the required sort data. On this screen you can set up a new structure type (if the current structure type is standard) or update data associated with the current structure.

P

Purge the current structure and restore the standard structure. CA Telon displays the Update Sort Definition screen, where you confirm the purge. You can also proceed to select the match sort or merge sort structure.

Z

Purge the current structure and all data associated with it, and restore the standard structure.

MERGE

A field in which a plus sign (+) indicates that merge sort is the current structure of the program. You can overwrite + with one of these values:

U

Transfer to the List Merge Key Groups screen to enter the required sort data. On this screen you can set up a new structure type (if the current structure type is standard) or update data associated with the current structure.

P

Purge the current structure and restore the standard structure. CA Telon displays the List Merge Key Groups screen, where you confirm the purge. (You can then proceed to select the mainline sort or match sort structure.)

Z

Purge the current structure and all data associated with it, and restore the standard structure.

MATCH

A field in which a plus sign (+) indicates that the program contains master/transaction match processing. You can over type + with one of these values:

U

Transfer to the Update Match Keys screen to enter the required sort data. On this screen you can set up a new structure type (if the current structure type is standard) or update data associated with the current structure.

P

Purge the current structure and restore the standard structure. CA Telon displays the Update Match Keys screen, where you confirm the purge. You can also proceed to select the mainline sort or merge sort structure.

Z

Purge the current structure and all data associated with it, and restore the standard structure.

APPLID

The application ID. The default is the value entered in the APPLID field on the Update Program Definition Defaults screen.

The use of the application ID is defined during CA Telon installation. See your system administrator for details.

USER SORTS

A field in which a plus sign (+) indicates that the program contains a user sort. You can overtype + with one of these values:

U

Transfer to the List Sorts screen to enter the required sort data.

P

Purge the current structure and restore the standard structure. CA Telon displays the List Sorts screen, where you confirm the purge.

CMPLOPT

Compiler parameters to be included in the generated program before the COBOL IDENTIFICATION DIVISION line or the PL/I PROC statement. The field on this screen contains 16 bytes. Once an entry has been made in the field, an extension field is presented after the field. If you need to enter a longer value, place a "U" in the extension field to go to the "Update Parameter Overflow" screen, where you can enter a total of 253 bytes, including commas.

IDENTIF

The custom code COPY member name to be added after the COBOL IDENTIFICATION DIVISION line for specification of INITIAL and other Identification Division options, or in the parentheses after OPTIONS in the PL/I PROC statement.

Note: When this copybook is used for PL/I, "MAIN" must be coded if it is desired.

COBFCPY (COBOL Only)

Two copy member names placed in the ENVIRONMENT DIVISION and DATA DIVISION of the program, respectively, specified in the SELECT and FILEDEF fields, respectively.

PROCEDR

The custom code COPY member name to be added before the PROCEDURE DIVISION line for specification of Declaratives after the Procedure Division. This copybook is valid only for COBOL; it does not appear on the screen for a PL/I program.

RPTDEST

The destination of report output (if any). Values include:

- The DDNAME of a sequential file
- A GSAM DBD
- The DDNAME of a VSAM ESDS data set

The default value is REPORT. The value must be defined in the data group for this batch definition. Otherwise, the FD OPEN and CLOSE for the output report are not generated and must be handled in custom code.

SELECT

The member containing select statements for the FILE-CONTROL section of the program.

FILEDEF

The member containing user-defined file definitions for the FILE section of the program.

WKAREA

The name of each copy member that contains the definition of a work area added to the DATA DIVISION section of the COBOL program.

There is a maximum string length of 253 bytes.

You can specify as many as 20 member names. Separate member names with commas.

You can include the members as part of the batch definition or as members of a library.

INIT1

Custom code placed in the Q-1000 section, after the R-1000 section and before open files. Use this code to perform any setup necessary before opening files, cursors, and so forth.

INIT2

Custom code placed in the Q-1000-PROGRAM-INIT section of the generated program. It is executed after the program opens the files.

GETTRAN

Custom code placed in the C-1000-GET-TRAN section of the generated program, which controls the retrieval of a batch input transaction.

It is performed first after program initialization and then from the main process loop after the printing of each detail. Once a valid transaction has been retrieved, A-1000-PROCESS-TRAN is performed to process the retrieved transaction.

This field and its select field and label are protected and blank for any match sort programs or mainline sort programs with an input procedure, since this custom code point is not used in those programs.

INMAST

(Match sort structure only.) Custom code placed in the C-1000-GET-MAST section of the generated program. Use it for processing immediately after a master record fetch.

Select, label, and name fields remain blank and protected until you select the program structure in the MATCH field on this screen.

INTRAN

(Match sort structure only.) Custom code placed in the C-1000-GET-TRAN section of the generated program. Use it for processing immediately after a transaction fetch.

Select, label, and name fields remain blank and protected until you select the program structure in the MATCH field on this screen.

ENDTRAN

(Match sort structure only.) Custom code placed in the C-1000-TRAN-DONE section of the generated program. Use it for processing after all transactions have been fetched for a single key.

Select, label, and name fields remain blank and protected until you select the program structure in the MATCH field on this screen.

PRCTRAN

Custom Code placed in the A-1000-PROCESS-TRAN section of the program, which processes a transaction retrieved in C-1000-GET-TRANSACTION. By default, this routine selects the first detail group in the batch definition printed into the report. Selection of other detail groups must be done using the custom code specified here.

This field and its select field and label are protected and blank for any match sort programs or mainline sort programs with an input procedure, since this custom code point is not used in those programs.

MATCH

(Match sort structure only.) Custom code placed in the A-1000-MAST-EQ-TRAN section of the generated program. Use it for processing when the master and transaction keys are equal.

Select, label, and name fields remain blank and protected until you select the program structure in the MATCH field on this screen.

Note: This custom code is invoked from A-1000-MATCH.

MGREATR

(Match sort structure only.) Custom code placed in the A-1000-MAST-MGREATER section of the generated program. Use it for processing when the master key is greater than the transaction key.

Select, label, and name fields remain blank and protected until you select the program structure in the MATCH field on this screen.

Note: This custom code is invoked from A-1000-MATCH.

TGREATR

(Match sort structure only.) Custom code placed in the A-1000-TRAN-TGREATER section of the generated program. Use it for processing when the transaction key is greater than the master key.

Select, label, and name fields remain blank and protected until you select the program structure in the MATCH field on this screen.

Note: This custom code is invoked from A-1000-MATCH.

TERM

Custom code placed in the T-1000-PROGRAM-TERM section of the program before closing the batch files.

SECTION

Custom code COPY or INCLUDE members added as COBOL sections or PL/I procedures, performed from other parts of the program. Specify a maximum of 35 members, each separated with a comma.

There is a maximum string length of 253 bytes.

The members can be located in a common library or can be unique to this program.

PGMCUST

The name of the COBOL section or PL/I procedure in which to add custom code, and the name of the custom code member added.

There is a maximum string length of 253 bytes.

You can make multiple specifications using this format:

section-name1, member-name1,
section-name2, member-name2,...

Section-name

The four-character identifier of the section or procedure in which to include the custom code (for example, H100) and a suffix (I or T) that specifies whether to include the code at the beginning (I) of the section or procedure, or at the end (T).

For example, H100I specifies that section H100 is included at the beginning of the program and E100T specifies that section E100 is included at the end of the program.

Member-name

The name of the custom code added at the location specified by *section-name*.

Thus, the value A100I,OUTIDC specifies that the custom code named OUTIDC is placed at the beginning of the A-100 section.

The following section names are available for Batch programs:

Section	Description
A1000I	A-1000-PROCESS-TRAN, A-1000-MATCH (beginning of section; batch standard, batch match, respectively)
A1000T	A-1000-PROCESS-TRAN, A-1000-MATCH (end of section; batch standard, batch match, respectively)

Section	Description
A1000_2I	A_1000_TRAN_GREATER (beginning of section; PL/I batch match)
A1000_3I	A-1000-MAST-GREATER (beginning of section; PL/I batch match)
A1000_4I	A-1000-MAST-EQUAL-TRAN (beginning of section; PL/I batch match)
A1000-2I	A-1000-TRAN-GREATER (beginning of section; COBOL batch match)
A1000-3I	A-1000-MAST-GREATER (beginning of section; COBOL batch match)
A1000-4I	A-1000-MAST-EQUAL-TRAN (beginning of section; COBOL batch match)
B1000I	B-1000-PROCESS-DETAIL (beginning of section)
B1000T	B-1000-PROCESS-DETAIL (end of section)
B2000I	B-2000-END-REPORT (beginning of section)
B2000T	B-2000-END-REPORT (end of section)
B5000I	B-5000-FORMAT-<groupname> (beginning of every B-5000-FORMAT-<groupname> section)
B5000T	B-5000-FORMAT-<groupname> (end of every B-5000-FORMAT-<groupname> section)
B6000I	B-6000-PRINT-<groupname> (beginning of every B-6000-PRINT-<groupname> section)
B6000T	B-6000-PRINT-<groupname> (end of every B-6000-PRINT-<groupname> section)
B9000I	B-9000-PAGE-BREAK (beginning of section)
B9000T	B-9000-PAGE-BREAK (end of section)
C1000I	C-1000-GET-TRAN (beginning of section)
C1000T	C-1000-GET-TRAN (end of section)
C1000_2I	C_1000_FIND_MIN_KEY (beginning of section; PL/I batch match)
C1000_2T	C_1000_FIND_MIN_KEY (end of section; PL/I batch match)

Section	Description
C1000_3I	C_1000_CHECK_KEY_LEVELS (beginning of section; PL/I batch merge)
C1000_3T	C_1000_CHECK_KEY_LEVELS (end of section; PL/I batch merge)
C1000_4I	C_1000_SET_INDICATORS (beginning of section; PL/I batch merge)
C1000_4T	C_1000_SET_INDICATORS (end of section; PL/I batch merge)
C1000-2I	C-1000-FIND-MIN-KEY (beginning of section; PL/I batch merge)
C1000-2T	C-1000-FIND-MIN-KEY (end of section; PL/I batch merge)
C1000-3I	C-1000-CHECK-KEY-LEVELS (beginning of section; COBOL batch merge)
C1000-3T	C-1000-CHECK-KEY-LEVELS (end of section; COBOL batch merge)
C1000-4I	C-1000-SET-INDICATORS (beginning of section; COBOL batch merge)
C1000-4T	C-1000-SET-INDICATORS (end of section; COBOL batch merge)
C2000I	C-2000-WRITE-REPORT (beginning of section)
C2000T	C-2000-WRITE-REPORT (end of section)
MAINI	MAIN (beginning of section)
MAINT	MAIN (end of section)
MAININLOOP	MAIN_INPUT_LOOP (beginning of section; PL/I batch)
MAININLOOPE	MAIN_INPUT_LOOP (end of section; PL/I batch)
MAINLOOP	MAIN_PROCESS_LOOP (beginning of procedure; PL/I batch only)
MAINOUTLOOP	MAIN_OUTPUT_LOOP (beginning of procedure; PL/I batch only)
MAINOUTLOOPE	MAIN_OUTPUT_LOOP (end of procedure; PL/I batch only)
MAINPROCESSI	MAIN-PROCESS (beginning of section)
MAINPROCESST	MAIN-PROCESS (end of section)

Section	Description
Q1000I	Q-1000-PROGRAM-INIT (beginning of section)
Q1000T	Q-1000-PROGRAM-INIT (end of section)
R1000I	R-1000-PARSE-PARM (beginning of section)
R1000T	R-1000-PARSE-PARM (end of section)
S1000I	S-1000-USER-SORT (beginning of section)
S1000T	S-1000-USER-SORT (end of section)
SRTINPUTLOOP	MAIN-INPUT-LOOP (beginning of section; batch mainline sort)
SRTINPUTLPE	MAIN-INPUT-LOOP (end of section; batch mainline sort)
SRTOUTPUTLOOP	MAIN-OUTPUT-LOOP (beginning of section; batch mainline sort)
SRTOUTPUTLPE	MAIN-OUTPUT-LOOP (end of section; batch mainline sort)
T1000I	T-1000-PROGRAM-TERM (beginning of section)
T1000T	T-1000-PROGRAM-TERM (end of section)
Z100I	Z-100-SECTIONS-COPY (beginning of section)
Z900I	Z-900-SECTION-FALLOUT & Z-900-PROGRAM-END (beginning of section); COBOL programs only)
Z970I	Z-970-IDMSSQL-ERROR (beginning of section)
Z970T	Z-970-IDMSSQL-ERROR (end of section)
Z980I	Z-980-ABNORMAL-TERMINATION (beginning of section)
Z980T	Z-980-ABNORMAL-TERMINATION (end of section)
Z990I	Z-990-PROGRAM-ERROR (beginning of section)
Z990T	Z-990-PROGRAM-ERROR (end of section)

PARMS

The length of each parameter field parsed from the JCL parameter passed to the batch program. The number of parameters is equal to the number of lengths you enter here.

Fields are parsed by commas, quotes, and parentheses, and placed in generated fields with lengths corresponding to the length values you specify.

Update Batch Environment

Access

On the Batch Program Definition menu, enter:

- **UP** in the FUNCTION field
- **EN** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field

Program ID

B168

Function

Specifies required information about environmental characteristics under which the generated batch program operates.

```

XXXXXX.BD UPDATE      _____ BATCH ENV *****
COMMAND ==> _____

GENERAL: TRACE        (Y/N)
          DBMS
          * PGMNAME    _____

LINKAGE: LNKCOPY      _____
          * USGCOPY    _____

DLI:    GENPCBS        (Y/N)
          * DLIWGHT     (Y/N)
          * PSBNAME     _____

PL/I:    PLIXOPT _ (C-CREATE/U-UPDATE/P-PURGE)
  
```

Field Definitions

COMMAND

For information, see Primary Commands.

TRACE

A value to specify whether CA Telon is to generate and maintain trace variables to provide added information that facilitates the testing and debugging process. Values are:

Y

(Default) Generate and maintain trace variables in the program

N

Do not generate and maintain trace variables in the program

DBMS

The database management system. For z/OS batch, values are:

- DL/I
- EXEC DLI
- VSAM
- SEQ
- DB2
- IDMS SQL (SQL)
- DATACOM (SQL)

PGMNAME

The name of the load module created by the linkage editor, required only when the load module name is different from the name that CA Telon generates for the program.

You specify this name in the format *hhxxnnnn*, where:

hh

The header

xx

One or two constant characters as specified in the PGMNAMES macro by target environment (for example, batch, IMS, CICS)

nnnn

The identifier

For example, *hhBPnnnn* is the default for a COBOL batch program.

LNKCOPY

The name of the COPY member containing required PCB definitions. These are required 01-level declarations included in the linkage section of the COBOL batch program.

USGCOPY

The list of variable declarations in the linkage member appended to the COBOL PROCEDURE DIVISION or the PL/I procedure statements.

GENPCBS

A value to specify whether CA Telon is to generate DL/I PCB masks in the generated program. Values are:

Y

Automatically generate PCB masks in the program.

N

Do not automatically generate PCB masks. In this case, the programmer must supply PCB masks in the LNKCOPY and USGCOPY custom code members.

Note: If the DBMS field value is EXEC DLI, CA Telon expects the GENPCBS value to be N. If you have included the I/O PCB in data access, you are responsible for coding that PCB as well as any other PCBs. You should define it as IO-PCB (or IO_PCB for PL/I) to avoid compiler errors.

DLIWGHT

A value to specify whether the generated program automatically weight DL/I calls. Values are:

Y

Automatically weigh DL/I calls

N

Do not automatically weigh DL/I calls

If you enter **Y**, the variable DLI-ACCUM-WEIGHT is generated. During execution it is incremented by 3 for each DL/I ISRT or DLET and by 1 for any other call. This allows you to determine when to take a checkpoint. Custom code is required to check and reset this variable.

PSBNAME

The PSBNAME for the DLIPSB statement for this batch program.

PLIXOPT (PL/I only)

A field in which a plus sign (+) indicates the existence of a PLEXOPT statement for this environment. You can overwrite + with one of these values:

C

Create a PLEXOPT statement for this environment

U

Update an existing PLEXOPT statement for this environment

P

Purge an existing PLEXOPT statement for this environment

Field Definitions

COMMAND

For information, see Primary Commands.

MASTER FILE

(Protected field.) Displays the name of the match master file identified on the Create/Update Data Group screen when MATCHM is specified in the REQUEST field.

TRANSACTION FILE

(Protected field.) Displays the name of the match transaction file identified on the Create/Update Data Group screen when MATCHT is specified in the REQUEST field.

SEQ

A field in which to enter line commands that manipulate the sequence key groups. Values are:

A

After

B

Before

D

Delete

X

Exclude

F

Show first excluded line

I

Insert

L

Show last excluded line

M

Move

See Line Commands for more information.

DATANAME

Key field for a master file or a transaction field.

The screen provides space for 15 keys for the master file and 15 keys for the transaction file. You must identify at least one key field for the master and transaction files, and you must enter the same number of keys for each file.

MORE

A field in which to request to transfer to the Update Parameter Overflow screen to enter a DATANAME value longer than the 25 characters permitted on this screen, increasing the maximum to 60 characters. To request transfer, enter a nonblank character.

LTH

The length of the match key field, in bytes. A value is required here when you enter a DATANAME field value.

This value is used with the values in the FORM and SIZE fields to generate a save area for each file's match keys.

FORM

The key field group's format. Values are:

A

(Default) Alphanumeric

N

Numeric

SIZE

Two fields defining the size of the key field when the FORM value is N (numeric). The first field specifies the number of positions to the right of the decimal point; the second field specifies the number of positions to the left of the decimal point.

The sum of the SIZE fields must equal the value in the LTH field.

List Sorts

Access

On the Create/Update Batch Definitions screen, enter **U** in the USER SORTS field.

Program ID

B1S1

Function

Lists up to 20 usersorts (and the mainline sort, for a mainline sort program) that are included in a program, and allows access to the Update Sort Definition screen where you can enter sort specification details for a selected sort.

hhtttt.BD LIST USER SORTS *****		
COMMAND ==>		
*	SORTNAME	DESCRIPTION
-	-----	-----
-	-----	-----
-	-----	-----
-	-----	-----
-	-----	-----
-	-----	-----
-	-----	-----
-	-----	-----
-	-----	-----
-	-----	-----
-	-----	-----
-	-----	-----
-	-----	-----
-	-----	-----
-	-----	-----
-	-----	-----
-	-----	-----
-	-----	-----
-	-----	-----
-	-----	-----
-	-----	-----

Field Definitions

COMMAND

For information, see Primary Commands.
You can also specify **CREATE** *sortname* ["*description*"].

(Line Command)

A field in which to enter one of these line commands:

U

Update the sort on the Update Sort Definition screen, where the full sort specifications are made

C

Copy an existing sort, with all its data

P

Delete the sort

Z

Delete the sort immediately

S

Show the Sort Definition screen

Note: For a mainline sort, only U and S line commands are valid.

SORTNAME

The name to identify a sort, used in generating an S-1000-*sortname* section.

The value MAINLINE is not valid because it is reserved for the mainline sort. If a mainline sort program has user-defined sorts, the mainline sort is listed first on the screen. You can access the mainline sort in the same manner used to access user-defined sorts, but to alter or delete it, you must go to the Create/Update Batch Definitions screen.

DESCRIPTION

(Informational only.) Description of the sort. It is stored in the TDF database, but is not used in code generation. You can update it by typing over it.

List Merge Key Groups

Access

On the Create/Update Batch Definitions screen, enter **U**, **S**, or **P** in the MERGE field of the STRUCTURE line.

You can enter **S** in the same field on the List Merge Key Groups screen to display this screen with all fields except the COMMAND line protected.

Program ID

B1M1

D

Delete

X

Exclude

F

Show first excluded line

I

Insert

L

Show last excluded line

M

Move

R

Repeat

U

Update

See Line Commands for more information.

LTH

The length of the merge key field, in bytes. This value is used with the values in the FORM and SIZE fields to generate a save area for each file's merge keys.

FORM

The key field group's format. Values are:

A

(Default) Alphanumeric

N

Numeric

SIZE

Two fields defining the size of the key field when the FORM value is N (numeric). The first field specifies the number of positions to the right of the decimal point; the second field specifies the number of positions to the left of the decimal point.

The sum of the SIZE fields must equal the value in the LTH field.

DESCRIPTION

(Informational only.) Description of the group. It is stored in the TDF database, but is not used in code generation. You can update it by overtyping it.

Update Sort Definition

Access

Access this screen in one of these ways:

- On the List Sorts screen, enter **U** as a line command for the listing of the sort to update
- On the Create/Update Batch Definitions screen, enter **U**, **S**, or **P** in the MAINSORT field
- On the Show/Purge Batch Definitions screen, enter **S** in the MAINSORT field

Program ID

B1S2

Function

Specifies the values needed to define a sort (mainline or user-defined).

Screen display for COBOL

[illegible]

COBOL programs require the following:

- Input proc (PROCIN field) or file (FILEIN field)
- Output proc (PROCOUT field) or file (FILEOUT field)
- An LRECL specification
- At least one set of the key fields

For a COBOL program containing a sort (either mainline or user-defined), one COBOL SELECT is built with the name SORTWK01. For COBOL programs, CA Telon builds a single COBOL SELECT statement for each file being sorted, regardless of the number of sorts each is used in.

For user-defined sorts, input and output PROC names must be different. It is the programmer's responsibility to release the sort record in the input PROC and return the sort record in the output PROC.

Screen display for PL/I

```

XXXXXX.BD  UPDATE SORT *****
COMMAND ==> _____ SCROLL ==> _____
SRTNAME _____ DESCRIPTION _____
DEFINITION:
*   PROCIN  _____
*   PROCOUT  _____
*   COLLATE  _____ PREFIX _____ STORAGE _____
FILE  LRECL  _____ (MIN MAX)
*   COPY    _____ COPYLV1 _ (Y/N)
*   COPYLBL  _____
FIELDS:
***** START LTH ORDER FORM
***** **** * * *
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
***** **** * * *

```

PL/I programs require:

- An LRECL specification
- At least one key field

If you do not enter PROCIN or PROCOUT values for a PL/I sort, CA Telon assumes the sort is FILEIN/FILEOUT (for example, PLISRTA).

If a PREFIX value is not entered, the FILEIN file name defaults to SORTIN and the FILEOUT name defaults to SORTOUT. If a PREFIX value is entered (for example, TASK), CA Telon adds the suffixes IN and OUT to the PREFIX value to create the file names (for example, TASKIN and TASKOUT).

If you enter a PROCIN but not a PROCOUT value for a PL/I sort, CA Telon assumes the sort is PROCIN/FILEOUT (for example, PLISRTB), and the FILEOUT name is built using the PREFIX value (either what you enter or the default value SORT).

If you enter a PROCOUT value but not a PROCIN value for a PL/I sort, CA Telon assumes the sort is FILEIN/PROCOUT (for example, PLISRTC), and the FILEIN name is built using the PREFIX value (either what you enter or the default value SORT).

If you enter both PROCIN and PROCOUT values for a PL/I sort, CA Telon assumes the sort is PROCIN/PROCOUT (for example, PLISRTD).

Field Definitions

COMMAND

For information, see Primary Commands.

SORTNAME

(Protected field.) Displays the name of the sort. Possible values are MAINLINE or, for a user-defined sort, the value entered in the SORTNAME field on the List Sorts screen.

PROCIN

A custom code member included in the sort's INPUT PROCEDURE.

FILEIN (COBOL Only)

A SORT USING (that is, with no processing to prepare input).

PROCOUT

A custom code member included in the sort's OUTPUT PROCEDURE.

FILEOUT (COBOL Only)

A SORT GIVING (that is, with no processing on output).

COLLATE

The collating sequence. Values are:

E

(Default) EBCDIC

A

ASCII

(name) (COBOL Only)

Custom code member containing a user-defined collating sequence

PREFIX (PL/I Only)

Input a four-character code used to construct names for USING and/or GIVING files that can be referenced before or after a sort (also SORTWK files).

The default prefix is SORT, which means the default name for a USING file is SORTIN and for a GIVING file is SORTOUT. However, if you enter a prefix value, such as TASK, references to an input file are to TASKIN (for a file-in sort), and to an output file are TASKOUT (for a file-out sort). For a procedure-in or procedure-out sort, no file references are generated.

STORAGE (PL/I Only)

A numeric value equaling the amount of main storage to be made available to the sort program. Minimum allowed is 88K (90112). If not specified here, the default is handled by generating the following code (which uses the maximum storage available):

```
DCL MAXSTOR    FIXED BINARY(31,0);
UNSPEC(MAXSTOR) = '00000000'B || UNSPEC ('MAX');

.
.
.
CALL PLISRTX
  (...
    ...
    MAXSTOR ...
```

About FILE fields

The four FILE fields are LRECL, COPY, COPYLV1, and COPYLBL. These fields define the sort file used for input and/or output procedure sorts. In COBOL sort programs, a sort COBOL SELECT statement is built for the sort file based on the FILE fields; in PL/I programs, the information is used to build the sort statement and input and/or output procedures.

LRECL

The maximum and minimum record size of the sort file used for sorts with both input and output procedures.

For sorts with an input or output file, record size is taken from the file's data group definition. For sorts with both input and output files, the LRECL of values of the files must match.

COPY

The name of the COPY member containing the layout of the sort file used for input and/or output procedure sorts. If this value is used for COBOL, the sort field must be identified in the DATANAME field. Sort fields **must not** be defined positionally.

COPYLV1

A value to specify that the layout of the sort file specified in the COPYLBL field begins with a 01 level. If you specify a COPYLBL value, enter **Y** here.

COPYLBL

The I/O area name of the sort file used for input and/or output procedure sorts. This value overrides any existing specification.

About FIELDS fields

The FIELDS fields are SEQ, START, LTH, ORDER, FORM, and (COBOL only) SIZE and DATANAME. Use the appropriate combination of values in these fields to define a maximum of five fields per sort key.

For both PL/I and COBOL, values are required in the START, LENGTH, FORM, and ORDER fields for each key field. Alternatively, for COBOL, you can specify no values in these fields but instead enter a COPY member name in the DATANAME field.

SEQ

A field in which you can enter one of these line commands to manipulate the sequence of sort keys listed:

A

After

B

Before

D

Delete

X

Exclude

F

Show first excluded line

I

Insert

L

Show last excluded line

M

Move

See Line Commands for more information.

START

The starting byte for the sort key.

LTH

The length of the sort key, in bytes.

ORDER

The sorting sequence for the field. Values are:

A

(Default) Ascending

D

Descending

FORM

The format of the sort field. Values are:

CH

(Default) Character

ZD

Picture/zoned decimal

PD

Packed decimal

BI

Binary

FI

Fixed point

FL

Floating point

SIZE (COBOL Only)

Two fields defining the size of the key field when its type is ZD or FI. The first field specifies the number of positions to the right of the decimal point; the second field specifies the number of positions to the left of the decimal point.

The sum of the SIZE fields must equal the value in the LTH field.

DATANM (COBOL Only)

The name of the field to be sorted.

MORE

A field in which to request to transfer to the Update Parameter Overflow screen to enter a DATANAME value longer than the 25 characters permitted on this screen, increasing the maximum to 60 characters. To request transfer, enter a nonblank character.

DATA SET

(Protected field.) Displays the name of the merge data set to which the key belongs, derived from the LABEL field on the Create/Update Data Group screen. Data Set names are shown in the merge order defined on the Create/Update Data Group screen; that is, the first data set has auto exec access MERGE01, the fifth, MERGE05, and so on.

DATANAME

The name of the host variable for the key field. The variable must be defined in the program.

A DATANAME value is required for each DATA SET value displayed on the screen.

Create/Update Stored Procedure Definition

Access

On the Batch Program Definition menu, enter:

- **CR** or **UP** in the FUNCTION field
- **SP** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field

Program ID

B210

Function

Specifies values required to create a stored procedure program, including:

- The name of the program work area custom code
- The names of other custom code members added to the program

To save your entries and return to the Batch Program Definition menu, press End.

```

XXXXXXXX.SP UPDATE STORED PROCEDURE *** *****
COMMAND ==>
OPTIONS ==> CUSTOM CODE _ DATA GROUP _ PARAMETERS _ ENV STORED _

GENERAL: DESC _____ A REMARKS _____
*   L ANGLVL _____ SPNAME _____ LANG __ (COB/PLI)
*   APPLID _____
*   CMPLOPT _____ A IDENTIF _____ A PROCEDR _____

FILES: A COBFCPY:SELECT _____ FILEDEF _____

AREAS: A WKAREA _____

Q-1000 A INIT1 _____ A INIT2 _____
A-1000 A SPINIT _____
C-3000 A SPPROC _____
D-3000 A SPTERM _____
T-1000 A TERM _____

MISC: A SECTION _____
*   PGM CUST _____
*   RUNOPTS _____

```

Show/Purge screen

You can access the Show/Purge Stored Procedure Definitions (B214) screen from the Online Program Definition menu by entering:

- **SH** or **PU** in the FUNCTION field
- **BD** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field

Alternatively, on the List Panel Definitions screen, you can enter **S** or **P** as a line command for a stored procedure definition (SP).

Field Definitions

The Show/Purge Stored Procedure Definitions screen fields are the same as the Create/Update Stored Procedure Definitions screen fields.

COMMAND

For information, see Primary Commands. You can also enter one of the following commands to invoke the custom code editor for a specified member:

- **CREATE** *member-name* □ *member-description*Ù
- **EDIT** *member-name* □ *member-description*Ù
- **UPDATE** *member-name*
- **SHOW** *member-name*

Note: *Member-description* is optional.

OPTIONS

The screen to which you transfer to complete the necessary specifications of the program definition.

The table below shows the options by field name and the associated screen to which control transfers. Select a field by entering a nonblank character.

OPTIONS	Field Screen to Which Control Transfers
CUSTOM CODE	List/Show Custom Code
DATA GROUP	Update Database Segment
PARAMETERS	Update Stored Procedure Parameters
ENV STORED	Update Stored Procedure Environment

DESC

The description entered on the Stored Procedure Program Definition menu. You can change the description here.

(EDIT FLAG FIELD) A

A one-byte field immediately succeeding several fields, as shown on the screen illustration. Values are presented next:

Value	Meaning	Remarks
0	Open	Allows you to rename the custom code member or associate it with another entry point. It erases any value displayed in the field's name field (this is also true for the name "***DFLT***"). This action only disconnects the association between this entry and

Value	Meaning	Remarks
		the custom code member. It does not delete the member's contents.
S	Show	Allows you to browse the associated member name.
U	Update	Allows you to update or create the member name. Not valid for purge or show modes. Open some members and show others.

REMARKS

(Protected field.) Identifies the name of the custom code member added in the COBOL REMARKS section of the program or at the beginning of the PL/I program.

LANGLVL

The CA Telon release version. For stored procedures, the default is the current release at your site.

SPNAME (display only)

The external name of the stored procedure. This is an output-only field, only displaying a value if the name exists. The external name may only be specified through use of the DB2 Stored Procedure Catalog Extract.

LANG

The programming language in which the definition is generated. This field is used only when both COBOL and PL/I options are installed with CA Telon. The value specified here overrides the default-entered LANG field on the Update Program Definition Defaults screen.

APPLID

The application ID. The default is the value entered in the APPLID field on the Update Program Definition Defaults screen. The use of the application ID is defined during CA Telon installation. See your system administrator for details.

CMPLOPT

Compiler parameters to be included in the generated program before the COBOL IDENTIFICATION DIVISION line or the PL/I PROC statement. The field on this screen contains 16 bytes. Once an entry has been made in the field, an extension field is presented after the field. If you need to enter a longer value, place a "U" in the extension field to go to the "Update Parameter Overflow" screen, where you can enter a total of 253 bytes, including commas.

IDENTIF

The custom code COPY member name to be added after the COBOL IDENTIFICATION DIVISION line for specification of INITIAL and other Identification Division options, or in the parentheses after OPTIONS in the PL/I PROC statement.

PROCEDR

The custom code COPY member name to be added before the PROCEDURE DIVISION line for specification of Declaratives after the Procedure Division. This copybook is valid only for COBOL; it does not appear on the screen for a PL/I program.

COBFCPY (COBOL Only)

Two copy member names placed in the ENVIRONMENT DIVISION and DATA DIVISION of the program, respectively, specified in the SELECT and FILEDEF fields, respectively.

SELECT

The member containing select statements for the FILE-CONTROL section of the program.

FILEDEF

The member containing user-defined file definitions for the FILE section of the program.

WKAREA

The name of each copy member that contains the definition of a work area added to the DATA DIVISION section of the COBOL program. There is a maximum string length of 253 bytes. You can specify as many as 20 member names. Separate member names with commas. You can include the members as part of the batch definition or as members of a library.

INIT1

Custom code placed in the Q-1000-PROGRAM-INIT section, before open files. Use this code to perform any setup necessary before opening files, if specified.

INIT2

Custom code placed in the Q-1000-PROGRAM-INIT section of the generated program. It is executed after the program opens the files, if specified.

SPINIT

Custom code placed in the A-1000-STORED-INIT section of the generated program, which controls the mapping of stored procedure parameters to program variables.

SPPROC

Custom code placed in the C-3000-STORED-PROCESS section of the generated program. Use it for processing immediately after a generated DECLARE CURSOR and OPEN of the cursor.

SPTERM

Custom code placed in the D-1000-STORED-TERM section of the generated program, which controls the mapping of program variables to stored procedure parameters.

TERM

Custom code placed in the T-1000-PROGRAM-TERM section of the program before closing any opened files.

SECTION

Custom code COPY or INCLUDE members added as COBOL sections or PL/I procedures, performed from other parts of the program. Specify a maximum of 35 members, each separated with a comma. There is a maximum string length of 253 bytes.

PGMCUST

The name of the COBOL section or PL/I procedure in which to add custom code, and the name of the custom code member added. There is a maximum string length of 253 bytes. You can make multiple specifications using this format:

section-name1, member-name1, section-name2, member-name2, ...

Section-name The four-character identifier of the section or procedure in which to include the custom code (for example, H100) and a suffix (I or T) that specifies whether to include the code at the beginning (I) of the section or procedure, or at the end (T).

For example, H100I specifies that section H100 is included at the beginning of the program and E100T specifies that section E100 is included at the end of the program.

Member-name The name of the custom code added at the location specified by *section-name*.

Thus, the value A100I,OUTIDC specifies that the custom code named OUTIDC is placed at the beginning of the A-100 section.

The maximum string length is 253 bytes.

The next section names are available for stored procedure programs:

Section	Description
A1000I	A-1000-STORED-INIT (beginning of section)
A1000T	A-1000-STORED-INIT (end of section)
C3000I	C-3000-STORED-PROCESS (beginning of section)
C3000T	C-3000-STORED-PROCESS (end of section)
D1000I	D-1000-STORED-TERM (beginning of section)
D1000T	D-1000-STORED-TERM (end of section)
MAINI	MAIN-PROCESS (beginning of section)
MAINT	MAIN (end of section)
MAINLINE	MAIN (replaces entire section)
MAINPROCESSI	MAIN-PROCESS (beginning of section)
MAINPROCESST	MAIN-PROCESS (end of section)
S100I	S-100-STP-CALLS (beginning of section)
S100T	S-100-STP-CALLS (end of section)
S200I	S-200-STP-CURSORS (beginning of section)
S200T	S-200-STP-CURSORS (end of section)
Z100I	Z-100-SECTIONS-COPY (beginning of section)
Z900I	Z-900-SECTION-FALLOUT & Z-900-PROGRAM-END (beginning of section); COBOL programs only)
Z980I	Z-980-ABNORMAL-TERMINATION (beginning of section)
Z980T	Z-980-ABNORMAL-TERMINATION (end of section)
Z990I	Z-990-PROGRAM-ERROR (beginning of section)
Z990T	Z-990-PROGRAM-ERROR (end of section)

RUNOPTS

A comma-separated list of stored procedure run options to be specified in the CREATE PROCEDURE statement generated for this program definition.

Update Stored Procedure Environment

Access

To access the stored procedure definition view of this update screen, on the Create/Update Stored Procedure screen, enter a nonblank character to select the ENV STORED Field.

You can access a view of this screen also by entering the following on the Batch Program Definition menu:

- **CR, UP, or SH** in the FUNCTION field
- **EN** in the ITEM field
- *Header* in the HEADER field
- *Identifier* in the ID field
- **STORED** in the ENVIRON field

Program ID

B268

Function

Specifies the environment data for the stored procedure definition.

```
HHNNNN.SP UPDATE STORED ENV *****
COMMAND ==> _____

      LANG
RESULTS _ ASUTIME ____ WLMENV _____
COLLID  _____ SCHEMA _____

PARAMETERS:
* EXTSCUR: _ (2-DB2 U-USER D-DEFINER)
* PRIMSTYL: _ (N-GENERAL WITH NULLS G-GENERAL D-DB2SQL J-JAVA)
* PROGTYP: _ (M-MAIN S-SUB)
* SQLMOD : (M-MODIFIES SQL DATA N-NO SQL S-CONTAINS SQL
            R-READS SQL DATA)
```

```
SELECT ALL THAT APPLY (Y/N):
_ COMRETN      _ DBINFO      _ DETERM
_ FENCED      _ NULCALL      _ STAYRES
```

Show/Purge screen

You can access the Show/Purge Stored Procedure Environment (B269) screen from the Batch Definition menu screen by entering:

- **SH** or **PU** in the FUNCTION field
- **EN** in the ITEM field
- *Name* in the NAME field
- *Identifier* in the ID field

Field Definitions

The Show/Purge Stored Procedure Environment screen fields are the same as the Update Stored Procedure Environment screen fields.

COMMAND

For information, see Primary Commands.

LANG

The language for the stored procedure (COB or PLI). This is an output-only field.

RESULTS

Specifies the maximum number of result sets that the stored procedure can return. The default is 0, which indicates that there are no result sets.

ASUTIME

Specifies the total amount of processor time, in CPU service units, that a single invocation of a stored procedure can run.

WLMENV

Identifies the MVS Workload Manager (WLM) environment in which the stored procedure is to run when the DB2 stored procedure address space is WLM-established. The name of the WLM environment is a long identifier.

Note: If a WLM ENVIRONMENT is not specified, the stored procedure runs in the default WLM-established stored procedure address space specified at installation time.

COLLID

Identifies the package collection to be used when the stored procedure is executed. This is the package collection into which the DBRM that is associated with the stored procedure is bound.

If COLLID is not specified, the package collection for the stored procedure is the same as the package collection of the calling program.

SCHEMA

Part of the qualified name for of the stored procedure.

The qualified form of stored procedure name is a short SQL identifier (the SCHEMA name) followed by a period and a long SQL identifier.

EXTSCUR

Specifies how the stored procedure interacts with an external security product, such as RACF, to control access to non-SQL resources.

2

The stored procedure does not require a special external security environment. This is the only valid choice when a Workload Manager environment is not specified (DB2).

U

An external security environment should be established for the stored procedure. Access is performed using the authorization ID of the user who invoked the stored procedure (USER).

D

An external security environment should be established for the stored procedure. Access is performed using the authorization ID of the stored procedure owner (DEFINER).

PRMSTYL

Identifies the linkage convention (parameter style) used to pass parameters to the stored procedure. All of the linkage conventions provide arguments to the stored procedure that contain the parameters specified on the CALL statement. Some of the linkage conventions pass additional arguments to the stored procedure that provides more information to the stored procedure.

D

(DB2SQL) In addition to the parameters on the CALL statement, the following arguments are also passed to the stored procedure:

- a null indicator for each parameter on the CALL statement;
- the SQLSTATE to be returned to DB2;
- the qualified name of the stored procedure;
- the specific name of the stored procedure;
- the SQL diagnostic string to be returned to DB2; and
- the DB2INFO structure, if DBINFO is specified.

G

(GENERAL) Only the parameters on the CALL statement are passed to the stored procedure. The parameters cannot be null.

N

In addition to the parameters on the CALL statement, another argument is passed to the stored procedure. The additional argument contains a vector of null indicators for each of the parameters on the CALL statement that enables the stored procedure to accept or return null parameter values (GENERAL WITH NULLS).

J

The stored procedure uses a convention for passing parameters that conforms to the Java and SQLJ specifications. INOUT and OUT parameters are passed as single-entry arrays. The DBINFO structure is not passed (JAVA).

PROGTYP

Specifies whether the stored procedure will run as a main routine or a subroutine.

S

The stored procedure will run as a subroutine (SUB).

M

The stored procedure will run as a main routine (MAIN).

SQLMOD

Indicates whether or not the stored procedure can execute SQL statements and, if so, what type it can execute.

N

The stored procedure cannot execute any SQL statements (NO SQL).

M

The stored procedure can execute any SQL statement except those statements that are not supported in any stored procedure (MODIFIES SQL DATA).

R

The stored procedure cannot execute SQL statements that modify data. SQL statements that are not supported in any stored procedure return a different error (READS SQL DATA).

S

The stored procedure cannot execute any SQL statements that read or modify data. SQL statements that are not supported in any stored procedure return a different error (CONTAINS SQL).

COMRETN

Indicates whether DB2 commits the transaction immediately on return from the stored procedure (COMMIT ON RETURN).

N

DB2 does not issue a COMMIT when the stored procedure returns.

Y

DB2 issues a COMMIT when the stored procedure returns if the following statements are true:

- The SQLCODE that is returned by the CALL statement is not negative.
- The stored procedure is not in a must-abort state.
- The COMMIT operation includes work that is performed by the calling application process and the stored procedure.

If the stored procedure returns result sets, the cursors that are associated with the result sets must have been defined as WITH HOLD to be usable after the COMMIT.

DBINFO

Specifies whether specific information known by DB2 is passed to the stored procedure when it is invoked.

N

Additional information is not passed.

Y

An additional argument is passed to the stored procedure when it is invoked. This argument is a structure that contains information such as the application run-time authorization ID, the schema name, the name of a table or column that the procedure might be inserting into or updating, and identification of the database server that invoked the procedure.

DETERM

Specifies whether the stored procedure returns the same result from successive calls with identical input arguments.

Y

The stored procedure returns the same result from successive calls with identical input arguments (DETERMINISTIC).

N

The stored procedure might not return the same result from successive calls with identical input arguments (NOT DETERMINISTIC).

FENCED

Specifies whether or not the stored procedure runs in an external address space (to prevent user programs from corrupting DB2 storage).

Y

The stored procedure runs in an external address space (FENCED).

N

The stored procedure does not run in an external address space (NOT FENCED).

NULCALL

Determines whether or not the stored procedure will be called even when any of the input arguments is null, making the procedure responsible for testing for null argument values.

Y

The stored procedure will be called even when any of the input arguments is null; the stored procedure is responsible for testing for null argument values.

N

The stored procedure will not be call when any of the input arguments is null.

STAYRES

Specifies whether the stored procedure load module remains resident in memory when the stored procedure ends.

Y

The load module remains resident in memory after the stored procedure ends.

N

The load module is deleted from memory after the stored procedure ends.

Field Definitions

COMMAND

For more information, see Primary Commands.

(Line command) A

A field used to sequence fields using one of the following line commands:

- A—After
- B—Before
- C—Copy
- D—Delete
- X—Exclude
- F—Show first excluded line
- I—Insert
- L—Show last excluded line
- M—Move
- R—Repeat
- U—Update

For more information, see Line Commands.

NAME

The name of the parameter.

KIND

Describes the direction of the parameter to/from the calling program.
Values are:

Value	Meaning	Description
O	OUT	Output from stored procedure into calling program
I	IN	Input into stored procedure from calling program
B	INOUT	Both Input into and Output from the stored procedure

DBNAME

The variable name to or from which the stored procedure parameter is mapped. If the DBNAME is too long to fit on this screen, then the PF10 key may be used to toggle to the B2P2 program, which allows DBNAME specifications up to 60-characters in length.

EXTENSION (output only)

When the DBNAME is longer than 25 characters, and the B2P2 program has been used to specify a long DBNAME, then this field is displayed with a plus (+) sign, and the DBNAME field is protected.

(plus sign) B

A plus-sign in this field indicates that the parameter has a DBNAME that exceeds the 25 characters that may be entered on the screen. To enter or update a DBNAME with more than 25 characters, press PF10 to go to the Stored Proc Parm DBNAME Ext (B2P2) screen.

TYPE

Specifies the parameter type. These values result in generation of a COBOL or PL/I declaration, as follows (lth = user-supplied length; dec = user-supplied scale):

Type	SQL Type	COBOL	PL/I
DATE	DATE	X(10)	CHAR(10)
CHAR	CHAR, CHARACTER	X(lth)	X(lth)
DEC	DEC, DECIMAL (NUMERIC)	9(lth)V9(dec)	DEC FIXED(lth,dec)
FL	FLOAT	COMP-2	BIN FLOAT(53)
GR	GRAPHIC	G(lth/2)	GRAPHIC(lth/2)
INT	INT, INTEGER	S9(9) COMP	BIN FIXED(31)
STMP	TIMESTAMP	X(26)	CHAR(26)
NUM	NUMERIC (DEC, DECIMAL)	9(lth)V9(dec)	DEC FIXED(lth,dec)
DBLP	DOUBLE, DOUBLE PRECISION	COMP-2	BIN FLOAT(53)
REAL	REAL	COMP-1	BIN FLOAT(31)
SINT	SMALLINT	S9(4) COMP	BIN FIXED(15)
TIME	TIME	X(8)	CHAR(8)
VCHR	VARCHAR, CHAR VARYING	*X(lth)	CHAR(lth)
VGR	VARGRAPH, VARGRAPHIC	*G(lth/2)	GRAPHIC(lth/2) VAR

Note: * includes S9(4) COMP length variable

LTH

The length of the stored procedure parameter, which is used to the declaration for the parm. This field is required for the following parameter types (lth = user-supplied length; dec = user-supplied scale):

SQL Type	COBOL	PL/I
CHAR, CHARACTER	X(lth)	CHAR(lth)
DEC, DECIMAL (NUMERIC)	9(lth)V9(dec)	DEC FIXED(lth,dec)
GRAPHIC	G(lth/2)	GRAPHIC(lth/2)
NUMERIC (DEC, DECIMAL)	9(lth)V9(dec)	DEC FIXED(lth,dec)
VARCHAR	CHAR VARYING	*X(lth) CHAR(lth)
VARGRAPH	VARGRAPHIC *G(lth/2)	GRAPHIC(lth/2) VAR

Note: * includes S9(4) COMP length variable of the parameter.

DEC

The scale of the parameter (if applicable). Used only for DECIMAL and NUMERIC parameters.

IND

If the IND value is set to "Y" (yes), a nullable indicator is generated for this stored procedure parameter, and used as a parameter in the calling sequence for the stored procedure.

Stored Procedure Parameter DBNAME Extension

Access

On the Update Stored Procedure Parameters screen, select the PF10 key.

Program ID

B2P2

Function

Allows specification of full 60-byte DBNAME mappings for stored procedure parameters.

[illegible]

Field Definitions

COMMAND

For information, see Primary Commands.

PARAMETER (output-only)

The name of the stored procedure parameters.

DBNAME

The DBNAME which is mapped to the stored procedure parameter. When all desired mappings have been specified, use PF10 to toggle back to the B2P1 program. The page forward and page backward keys may be used to reposition the screen, if there are more parameters to be displayed.

Chapter 8: Prototyping Facility

The prototyping facility shows the application user how screens actually appear during system execution and how screens interact with each other during execution.

This chapter documents the screens of the CA Telon prototyping facility.

Requirements

To use the prototyping facility, you must have one or more panel images, with or without panel definitions.

To simulate application execution, you must:

- Enter data
- Edit the data
- Transfer from screen to screen

You can edit the data by including FLDTYPES and SPECIAL EDITS.

Controlling screen flow

You control flow from screen to screen in one of these ways:

- By entering the ID of next program in the COMMAND field
- By entering a panel ID as the value in one of the following:
 - NEXTPGM field of select fields on the Update Select Fields screen
 - INPUT or OUTIN fields, when NEXT-PROGRAM-NAME-ID is the DBNAME
 - NEXTPGM field on the Create/Update Screen Definition screen

Note: CA Telon does not support multiple prototyping facility sessions (using HOLD/SWAP). Use of HOLD for multiple prototyping facility sessions can produce unpredictable results.

Prototyping Facility Menu

This section discusses how to use the Prototyping Facility menu.

Access

On the TDF Main menu, enter **6** in the FUNCTION field. On other screens, enter **=6** in the COMMAND field.

Program ID

M100

Function

Displays the capabilities and features that you can use to perform panel prototyping.

```
PROTOTYPING FACILITY MENU *****
COMMAND ==> _____

FUNCTION: VI  VI-VIEW      LI-LIST
ITEM:      AP  AP-APPLIC  PS-PRESENTATION STORE

START NAME:
HEADER TR_
ID      CIDA_
PS NAME _____

ENTER DEFAULTS FOR INITIALIZING SCREEN FIELDS:
PRESENTATION: A + < 1 > (OUTIN INPUT SELECT OUTPUT CHARACTERS)
               B _ _ _ _ (OUTIFIL OVERRIDE - B=BLANK U=UNDERLINE N=NULL)
               C _ _ _ _ (UNDEFINED STORE CHARACTER)
               D _ _ _ _ (CONVERT INPUT TO UPPER CASE - Y/N)
               E _ _ _ _ (PROTOTYPE LEVEL - 1-WITHOUT DATA, 2 WITH DATA)
               F _ _ _ _ (FIELD EDIT AND FLOW EXECUTION - Y/N )

COMMAND POS:  __ _ (ROW - COLUMN OR G
                _____ NAME OF FIELD) H
                _ _ INTENSITY (N-NORMAL,B-BLANK,H-HIGH) I
```

Field Definitions

COMMAND

For more information, see Primary Commands in the chapter "Editors and Commands."

FUNCTION

The function performed. Values are:

VI

View an application or presentation store

LI

List application panels or presentation stores

ITEM

The item on which the specified function is performed.

This table presents valid ITEM values and the result, depending on the FUNCTION value:

FUNCTION	ITEM	Result
VI	AP	Begin an application scenario.
	PS	View a presentation store.
LI	AP	List application panels by header.
	PS	List existing presentation stores.

HEADER

(Valid when the ITEM value is AP.) A one- to five-character prefix for a series of screens for an application.

This table explains what the header identifies, depending on the FUNCTION field value:

FUNCTION	What the Header Identifies
VI	The application whose panels are viewed in the prototyping scenario.
LI	The application whose screens are listed.

Note: If you enter **LI** in the FUNCTION field and **AP** in the ITEM field, you must enter a HEADER value. If you do not, CA Telon displays an error message. Determine the appropriate header for your application before proceeding.

ID

(Valid when the ITEM value is AP.) A one- to five-character unique identifier for a screen in the series represented by the HEADER value.

This table explains what the ID value identifies, depending on the FUNCTION field value:

FUNCTION	What the ID Value Identifies
VI	The screen that the TDF is uses to begin the prototyping scenario.
LI	The starting point for which panels are listed.

If you do not enter a value here, a screen listing application panels starting with the lowest ID value of existing applications is displayed.

PS NAME

A one- to eight-character presentation store name for the prototyping facility to use with the scenario. This field is required only if you are using prototyping with mapping and a panel definition exists for the scenario.

A presentation stores represents sample data and is not necessarily tied to a specific panel or a specific application HEADER; therefore, CA Telon does not force a naming convention for presentation stores. However, recommended naming conventions are:

- Begin the name of a presentation store related to a single *panel* with the concatenation of the HEADER and ID values for the panel
- Begin the name of a presentation store related to a single *application header* with the application header
- Begin presentation stores that are versions of the same kind of data with a common prefix

PRESENTATION

Defaults for the prototyping facility session:

A

Enter the characters displayed on the screen for outin, input, select, and output fields, respectively. See Update Session Controls for CA Telon defaults.

Literals are automatically displayed.

B

Enter the character used to fill the outin, input, select, and output fields, respectively, on the application.

- B** Blank; fields are filled with spaces
- N** Null; fields are filled with low values
- U** Underline; fields are filled with underscores

C

Enter the undefined store character. CA Telon uses this character to fill an outin or output field when there is no DBNAME data value to display in the active presentation store.

If you change the undefined store character, the original undefined store character is used for fields defined before the change and the new character is used for fields defined after the change.

D

Character conversion specification. Values are:

- Y** Lower characters are converted to upper case characters during the prototyping session
- N** No character conversion occurs

This allows you to control how values entered are mapped to subsequent panels in the scenario.

E

The level of information used to build the panel scenario. Values are:

- 1** Only panel image information is used to build and display the panels, with the following results:

No editing takes place

Since no mapping can occur, the prototyping facility does not modify your active presentation store

The undefined store character does not appear

SEGLOOP screens show all occurrences of the SEGLOOP

You may use the exclamation point (!) character to access the field update screens

- 2** (*Default.*) Panel definition information is used to build the panels, with the following results:

Editing takes place

Since mapping can occur, the prototyping facility modifies your active presentation store

The undefined store character appears

You may use the exclamation point (!) character to access the field update screens

F

Method of editing and flow control. Values are:

Y All editing and flow control methods documented in data administration are available.

N Field editing is turned off. All flow control must be handled by the entries made in the COMMAND field or by .COMMAND-INIT/.COMMAND-FLOW DBNAMES in the presentation store.

See the &U\$TNPCT. for information about these entries.

COMMAND POS

The position of the command field and the intensity of its display:

G

Two fields specifying, respectively, the row and column position at which to begin the COMMAND field. The COMMAND field allows you to enter commands when you are prototyping. The default location is the lower-right corner of the View Panel Definition screen.

H

A panel field name from the existing panel definition (for example, ERRMSG1). The panel field name specifies where to place the COMMAND field and overrides a row and column position specified in the line above.

I

The intensity of the COMMAND field display:

B The field is invisible to the viewer but still accessible to enter commands

H High intensity using the outin character selected

N (*Default.*) Normal intensity using the outin character selected
enter commands

List Panel Definitions

Access

On the Prototyping Facility menu, enter:

- **LI** in the FUNCTION field
- **AP** in the ID field
- *Header* in the HEADER field
- *Identifier* in the ID field

Program ID

M401

Function

Lists application definition IDs by panel image or panel definition (PI/PD) name and allows you to view and maintain a panel.

Note: If a panel definition exists for a panel image, this screen lists only the panel definition.

LIST PANEL DEFINITIONS *****					
COMMAND ==>				PAGE 01	
A	PI/PD	PR STORE/			
	NAME	**RENAME**	*****DESCRIPTION*****	USER	UPDATE
	XXXXXX.PD		PANEL DEFINITION FOR DOCUMENTATION	XABCDEFG	082505
	YYYYYY.PD		PANEL DEFINITION FOR MODELING TEST	XYZAB02	091605
	ZZZZZZ.PD		PANEL DEFINITION FOR MODELING TEST	ABCDE04	112405

Field Definitions

COMMAND

For information, see Primary Commands.

Note: You can specify the starting point of the list by entering **L** *entity-name*. If *entity-name* is found, the list begins with *entity-name*. If *entity-name* is not found, the list begins with the name of the member that follows *entity-name* in alphabetical order.

(Line command) A

A field preceding each listed definition in which you can enter one of the commands documented in the following table:

Command	Function	Description
D	Description	Changes the description of the ID. Enter the new description in the DESCRIPTION field. When you press Enter, the screen is redisplayed with the new description and, in the RENAME field, a confirmation message: *DESC UP
P	Purge	Purges the ID with confirmation. When you press Enter, control passes to the Line Edit screen. CA Telon displays this message: ENTER 'PURGE' or 'PPPP' to delete member. Enter PURGE on the COMMAND line or, to cancel the purge and return the List Panel Definitions screen, enter CANCEL. Confirmation message: * PURGED. If you purge a panel definition, you must zap or purge the panel image separately. To list the panel image after returning to the List Panel Definitions screen, press Enter.
R	Rename	Renames the ID. Enter the new name in the RENAME field. When you press Enter, the screen is redisplayed with the new name and, in the RENAME field, a confirmation message: *RENAMED.
S	Show	Shows the panel image for the ID. When you press Enter, the screen is redisplayed in browse mode.
U	Update	Updates the panel image or panel definition of the ID. When you press Enter, the next screen is displayed in edit mode. If you request an update of a panel image, the next screen is the Edit Panel Image screen. If you request an update of a panel definition, the next screen is the Update Panel Fields (Online) screen. Confirmation message: *PNL SAVED.

Command	Function	Description
V	View	Displays the specified ID. When you press Enter, the next screen displayed is in edit mode. You enter the name of the presentation store in the RENAME field to begin a scenario. Confirmation message: *PNL VIEWED.
Z	Zap	Purges the ID without confirming the purge. If the panel definition is zapped, the panel image must also be zapped or purged separately.
C	Copy	Copies the ID. You must also make an entry in the DESCRIPTION field. When you press Enter, the screen is redisplayed with this confirmation message in the RENAME field: *COPIED

PI/PD NAME

(*Protected field.*) Displays the name of a panel definition or panel image.

PR STORE/RENAME

A field whose value is used depending on the line command entered for the listing:

Line Command	PR STORE/RENAME Value
V	The name of the presentation store used in the scenario. If the presentation store is to supply the panel with data, enter the value that is entered in the PS NAME field.
R	The new ID for the renamed definition. CA Telon adds .PI or .PD as appropriate to the value you enter and also displays confirmation message in this field.
C	The ID assigned to the copied definition. CA Telon adds .PI or .PD as appropriate to the value you enter and also displays confirmation message in this field.

DESCRIPTION

A 1- to 39-character description of the panel image or panel definition.

To change the displayed description of the panel, enter **D** in the line command field and enter a new description in this field.

USER

(*Protected field.*) Displays the ID of the user that last updated the definition.

UPDATE

(*Protected field.*) Displays in *mmddyy* format, the date the definition was last updated.

List Presentation Stores

Access

On the Prototyping Facility menu, enter:

- **LI** in the FUNCTION field
- **PS** in the ITEM field
- *Presentation-store* in the PS NAME field (optional)

Program ID

M200

Function

Lists presentation stores by presentation store name and allows you to maintain and edit them.

```

LIST PRESENTATION STORES ***** AT END OF HEADER REQUESTED
COMMAND ==> _____ PAGE 01
PI/PD
***NAME*** **RENAME** *****DESCRIPTION***** USER  UPDATE
MYPSTORE          ACTIVE PANEL AT CREATION WAS HHIIII.PD  XYZAB02  090705

```

Field Definitions

COMMAND

For information, see Primary Commands.

(Line command) A

A field preceding each listed presentation store in which you can enter one of the commands documented in this table:

Command	Function	Description
D	Description	Changes the description of the presentation store. Enter the new description in the DESCRIPTION field. When you press Enter, the screen is redisplayed with the new description and, in the RENAME field, a confirmation

Command	Function	Description
		message: *DESC UP
P	Purge	Purges the presentation store. When you press Enter, control passes to the presentation store editor. CA Telon displays this message: ENTER 'PURGE' or 'PPPP' to delete member. Enter PURGE on the COMMAND line or, to cancel the purge and return the List Panel Definitions screen, enter CANCEL . Confirmation message: * PURGED.
R	Rename	Renames the presentation store. Enter the new name in the RENAME field. When you press Enter, the screen is redisplayed with the new name and, in the RENAME field, a confirmation message: *RENAMED.
S	Show	Shows the presentation store via the presentation store editor. When you press Enter, the screen is redisplayed in browse mode.
U	Update	Updates the presentation store. When you press Enter, the next screen is displayed in edit mode. Confirmation message: *PS SAVED.
V	View	Same as Update. Confirmation message: *PS VIEWED.
Z	Zap	Purges the presentation store without confirming the purge.
C	Copy	Copies the presentation store. You may make an entry in the DESCRIPTION field or accept the base description, which you can modify later, if necessary. When you press Enter, the screen is redisplayed with this confirmation message in the RENAME field: *COPIED

NAME

(*Protected field.*) Displays the name of a presentation store.

RENAME

The new presentation store name when renamed (line command R) or copied (line command C). CA Telon also uses this field to display confirmation messages as presented in this table:

Line Command	Description	Confirmation Message
D	Changes the description of a presentation store	*DESC UP
P	Purges a presentation store and confirms the delete	*PURGED
R	Renames a presentation store	*RENAMED
S	Shows a presentation store	None
U	Updates a presentation store	*PS SAVE
V	Updates a presentation store	*PS SAVE
Z	Purges a presentation store without confirming the delete	None
C	Copies a presentation store	*COPIED

DESCRIPTION

A one- to 39-character description of the presentation store. A value is required if line command D or C is entered for this presentation store. The initial value is the ID of the panel for which the presentation store was created.

USER

(*Protected field.*) Displays the ID of the user that last updated the presentation store.

UPDATE

(*Protected field.*) Displays the date the presentation store was last updated.

View Presentation Store

Access

On the Prototyping Facility menu, enter:

- **VI** in the FUNCTION field
- **PS** in the ID field
- *Presentation-store* or blank (for the active presentation store) in the PS NAME field

You can also access this screen in one of these ways:

- On the View Panel Definition screen, enter **.VPS** in the command field (accesses the active presentation store only)
- On the List Presentation Stores screen, enter **U** or **V** as a line command for a specific presentation store

Program ID

M151

Function

Allows you to edit an active presentation store, a presentation store saved previously, or a newly created presentation store.

EDIT —	XXPSTR1	STORE 001 OF 001	SIZE 0003	COL 01
COMMAND ==>			SCROLL ==>	PAGE
A	***** VALUE *****	***** DBNAME *****	SUB	LTH
*****	***** TOP OF DATA *****	*****		
000001	RANDOM DATA VALUES	TEST-FIELD-1	001	030
000002	99	TEST-NUMERICS	001	002
000003	X	TEST-ALPHA	001	001

Adding a field

You can add a new field to the presentation store on this screen in one of these ways:

- On an empty line, enter VALUE and DBNAME values
- Enter **I** as a line command to insert. When inserting:
 - Enter the correct LTH value, if necessary (default is 30)
 - Enter the correct SUB value, if necessary (default is 1)

A presentation store may contain a maximum of 1000 named fields.

Field Definitions

COMMAND

For information, see Primary Commands.

Note: You can enter **EDIT** *presentation-store* to edit multiple presentation stores concurrently or to add a new store to the editing session.

Specify backward and forward paging in the COMMAND field to view multiple stores. If the presentation stores you are editing are small, you can display and edit them on the same screen. For editing, they are displayed as one entity, allowing you to copy variables across stores with line copy commands and change data in multiple stores with one CHANGE command.

When the edit session is complete, you can treat each store separately with commands such as *END presentation-store* and *CANCEL presentation-store*, or treat them as a group with commands such as *END ALL* and *CANCEL ALL*.

(Line command) A

A field in which you can enter standard line commands. See Line Commands for more information.

VALUE

A value supplied for a data field in the presentation store. You can modify, delete, or add values:

- To modify a value, type the new value over the old and press Enter.
- To delete a value, enter **D** on the line and press Enter.
- To add a value, type it in this field and press Enter.

If the length of a field is greater than the number of characters that can be keyed into the VALUE field, you can continue the VALUE entry into the next line by pressing Enter without entering DBNAME and LTH values.

DBNAME

The name of the field. Modifying this value does not change the DBNAME value on the panel definition.

When you enter a new DBNAME on a line with an existing value, CA Telon adds the new variable to the presentation store and maintains the specified VALUE.

These DBNAME values have special meaning to the prototyping facility. Only one of each may exist in a given presentation store:

- **.COMMAND-FLOW**— Permits you to transfer control to another application panel when that panel ID is in the corresponding VALUE field. **.COMMAND-FLOW** is executed only upon detection of a **..MRG** or **.LOD** command in the **COMMAND** field.

Note: **.LOD** or **..MRG**— can either be keyed into the **COMMAND** field or exist in the loaded or merged presentation store with **.COMMAND-INIT** as a **DBNAME**.

- **.COMMAND-INIT**— Permits you to issue **.LOD**, **.SAV**, etc. when the appropriate **COMMAND** field is in the corresponding **VALUE** field.

SUB

The subscript number of variables in an array. Subscripted variables are used in displaying multiple entries on a list screen or with non-list screens when the **.SUB** command has been used to set a subscript value. The default value for subscripts is 1 because unsubscripted variables are considered the same as subscripted variables with a subscript of 1.

Note: If multiple occurrences of a DBNAME have the same subscript, the presentation store editor ignores all but the last occurrence of the DBNAME.

LTH

The length of the value established in the panel definition. You can modify it here without changing the panel definition.

View Panel Definition

Access

On the Prototyping Facility menu, enter:

- **VI** in the FUNCTION field
- **AP** in the ID field
- *Header* in the HEADER field
- *Identifier* in the ID field

You can also access this screen from the List Panel Definitions screen by entering **V** as a line command for the listing of the definition.

Program ID

M120

Function

Displays the application scenario. You can enter data in the prototype for demonstration purposes. You can control the flow of the scenario, access the panel image and panel definition (if one exists), and execute other commands in the command field.

%%%%%%%%

TE L O N S A M P L E S O L U T I O N

EMPLOYEE %%%%

EMPLOYEE ID %%%%

1. NAME

2. STREET

3. CITY

4. STATE

5. ZIP

6. PHONE

7. SEX

8. DATE OF BIRTH

9. DEPARTMENT

10. DATE OF EMPLOYMENT

11. HOURLY RATE

12. HOURS PER WEEK

%%%

PFKEYS: 2-HOLD 3-END 4-ENDHOLD 5-CANCEL 6-MENU A_____

Field Definitions

(Command field) A

The field for entry of commands for:

- Screen flow control
- Prototyping commands
- TDF commands

The default position is the lower right corner of the screen.

Screen flow control

To transfer control from one panel to the next panel, identify the panel to receive control:

- When a panel is in the series represented by the current panel's header, enter the ID of the panel to be displayed next
- To call a panel under a different header, enter both the concatenation of the header and ID of the panel to be displayed next

For example, the header of the panel you are viewing is AB and the ID is 0001. To view panel AB0002, enter **0002** in the command field. To view panel 0001 in the DE series, enter **DE0001** in the command field.

Note: For panels having a panel definition that contains outin or input fields, press Enter. Data from the current screen is edited (if specified) and mapped to the active presentation store. To transfer to another panel without editing data or mapping to the active presentation store, prefix the panel ID with a greater than sign (>) (for example, >0001).

Prototyping commands

All commands begin with a period. This table presents commands you can use to control a prototyping session.

Command	Action
.CLR	Clears variable names and data from the active presentation store.
.CV	Clears values in the active presentation store, but does not clear the DBNAMEs .
.LOD <i>presentation-store</i>	Loads <i>presentation-store</i> and makes it the active presentation store. Data values in the prior active presentation store are lost.

Command	Action
.MRG <i>presentation-store</i>	Merges the presentation store name into the active presentation store. Replaces data values of similar DBNAMEs. Adds DBNAMEs and values.
.SAV <i>presentation-store</i>	Saves the active presentation store in <i>presentation-store</i> . If <i>presentation-store</i> already exists, it is overlaid. This operation does not affect values in the active presentation store.
.SUB <i>subscript</i>	<p>Sets the subscript for any array variables when used by non-list screens. The subscript is not used or affected by list screens. For non-list screens that use variables from an array, the .SUB command identifies which occurrence of the array is used. The subscript number entered stays in effect until a new .SUB command is entered.</p> <p>For example, if you enter .SUB 1, the subscript is cleared. Non-subscripted variables operate as if they are subscripted with (1). For example, if you enter .SUB 3, the third occurrence of any array variables is used for all panels executed until the next .SUB command.</p>
.UPI	Transfer control to the Edit Panel Image screen. When you press the END PF key, control returns to the application scenario from which you entered the .UPI command.
.UPD	Transfer control to the Update Panel Fields (Online) screen. When you press End, control returns to the application scenario from which you entered the .UPD command.
.VPS	<p>Transfer control to the presentation store editor to edit the active presentation store. From there, you can update, delete, or add any variable to the active presentation store. When you press the END PF key, control returns to the application scenario from which you entered the .VPS command.</p> <p>Note: The active presentation store is not permanently saved when you press End. Only the .SAV <i>presentation-store</i> command saves the active presentation store to a new <i>presentation-store</i> or replace an existing <i>ps-name</i>.</p>

TDF commands

This table presents the general TDF commands that are valid in the command field and are useful in a prototyping session.

Command	Action
/END	End prototyping facility execution and return to the Prototyping Facility menu
/MENU	Transfer control to the TDF Main menu
/HOLD	Hold execution and transfer to the TDF Main menu
/SWAP	Swap to another function executing concurrently through use of the HOLD command
/HELP	Request help for use of the prototyping facility function
/=n	End prototyping facility execution and transfer to TDF option <i>n</i> Note: These commands must begin with a forward slash.

Chapter 9: Utility Functions

With the CA Telon Design Facility (TDF) Utilities option you can copy, rename, list, print, and export these items created with the TDF:

- Panel image
- Panel definition
- Screen definition
- Report definition
- Driver definition
- Nonterminal definition
- Batch definition
- Stored procedure definition

Utilities Menu

Access

On the TDF Main menu, enter **U** in the FUNCTION field.

Program ID

U100

Function

Allows you to perform the following for each type of item:

- Create a copy of an item
- Rename an item
- List specific items and allow you to update, modify, or remove selected items
- Print the panel image (not available to TDF under CICS)
- Export a screen, report, driver, nonterminal, or batch definition into a CA Telon source code listing

```

UTILITIES MENU *****
COMMAND ==> _____

FUNCTION:  _ CO-COPY      RE-RENAME  LI-LIST      PR-PRINT
           _ XP-EXPORT    XR-EXPORT/REPLACE  LH-LIST HEADERS

MEMBER NAME:
HEADER  _____
ID      _____  TYPE  ____
DESC    _____

          (COPY/RENAME)  TO HEADER
                           ID      ____
                           _____  --

FORMATTING CONTROL: COMPRESSED  ____ (YES/NO)
                   ENVIRONMENT  ____ (C-CICS/I-IMS/T-TSO/B-BATCH/R-STORED
                                   S-CLIENT)
                   ENV FORMAT   ____ (YES/NO) (CICS:CICSBMS IMS:IMSMFS
                   ENV PSB      ____ (YES/NO) (CICS:DLIPSB  IMS:IMSPSB
                                   BATCH:DLIPSB)

PRINT CONTROL:  LINENUM  ____ (YES/NO)
                DBLSPC   ____ (YES/NO)
                SHIFT     ____ (YES/NO)
                USECOL1   ____ (YES/NO)
  
```

Field Definitions

COMMAND

For information, see Primary Commands.

FUNCTION

Utility function to perform. Values are:

CO

Copy the named member to the indicated header and ID. All lower levels are copied.

RE

Rename the named member to indicated header and ID. All lower levels are renamed.

LI

List members beginning with the member identified in the HEADER field (and, optionally, the ID field).

The list starts at the HEADER and ID you specify and continues to the end of the existing entities. It includes all panel images, panel definitions, screen definitions, driver definitions, and report definitions.

PR

Print a hard copy of the panel image identified by the HEADER and ID field values.

XP

Export the program definition identified by the HEADER and ID values. This creates a CA Telon source code listing of the named member that includes any custom code that is part of the program definition. The listing is stored in a partitioned data set (PDS). All lower levels are exported.

XR

The same function as XP except that it replaces an existing export stored in the PDS with a new export.

LH

List all headers currently in use in the TDF.

HEADER

(Required.) Enter the one- to five-character name prefixed to a series of programs in the same application. Length is determined at installation.

ID

The one- to five-character name used to identify a particular program within a header group.

TYPE

The item type to work with. Values are:

BD

Batch definition

DR

Driver definition

PD

Panel definition

PI

Panel Image

ND

Nonterminal definition

RD

Report definition

SD

(Default.) Screen definition

SP

Stored Procedure definition

DESC

(Informational only.) The header and ID description that identifies the member.

TO HEADER

New header for the item being copied (FUNCTION is CO) or renamed (FUNCTION is RE).

(TO) ID

New ID for the item being copied (FUNCTION is CO) or renamed (FUNCTION is RE).

COMPRESSED

Format of an exported member. Values are:

Y

Compress the format. Place as many fields as possible on each line.

N

(Default.) Do not compress the format. Place only one field per line (this improves readability).

ENVIRONMENT

Environment where the exported program runs after generation into a COBOL or PL/I program. Values are:

B

BATCH

C

CICS

I

IMS

S

CICS

T

TSO

ENV FORMAT (CICS and IMS Only)

Value used to specify whether the CA Telon Generator is to create IM MFS control blocks or a CICS BMS map. Values for CICS are:

Y

Create a BMS map during generation of the program.

N

(Default.) Do not create a BMS map. Use CA Telon's own terminal mapping facility (this is the recommended option when using 3270 type terminals).

Values for IMS are:

Y

Create IMS MFS control blocks during program generation. This creates the DIF/DOF, MID, and MOD control blocks. They are named according to conventions established at CA Telon installation.

N

(Default.) Do not create IMS MFS control blocks.

ENV PSB (CICS and IMS Only)

Value used to specify whether the CA Telon Generator is to create a PSB for the program. Values are:

Y

Create a PSB for the program. It is named according to conventions established at CA Telon installation.

N

(Default.) Do not create a PSB.

LINENUM (Print panel image only)

Value used to specify whether each panel image line should be numbered. The line numbers appear at each end of the print line and are separated from the panel image by asterisks. Values are:

Y

Print with line numbers

N

Do not print with line numbers

When the value is Y, JUXPRT prints in this format:

```
1 *      TRAINING SYSTEM      * 1
2 *          MENU              * 2
3 *                          * 3
```

DBLSPC (Print panel image only)

Value used to specify whether the numbered lines are printed double-spaced. A value here is meaningful only if the LINENUM value is Y. Values are:

Y

Double-space the printed lines (that is, insert a blank line, bordered by asterisks after each panel image line except the last)

N

Do not double-space the printed lines of the panel image

When the DBLSPC is Y and the LINENUM is Y, JUXPRT prints in this format:

```

      *
1 *      TRAINING SYSTEM      * 1
      *
2 *          MENU            * 2
      *
3 *                          * 3

```

SHIFT (Print panel image only)

Value used to specify whether to print column 1 of the panel image. A value here is meaningful only if the LINENUM value is N. Values are:

Y

Print column 1 characters of the panel image in column 2 of the output line

N

Print column 1 characters of the panel image in column 1 of the output line

Use this option for batch output in which column 1 contains values other than attribute bytes and carriage control characters.

USECOL1 (Print panel image only)

Value to specify, in conjunction with the value in the SHIFT field, whether to ignore screen control characters in column 1. Values are:

Y

Print any characters found in column 1; do not treat them as control characters

N

Set column 1 output to a space

List Headers

Access

On the Utilities menu, enter LH in the FUNCTION field.

Program ID

F144

Function

Allows you to list the application header in use in the TDF.

LIST APPLICATION HEADERS *****					
COMMAND ==> _____					PAGE __
U *HDR*	U *HDR*	U *HDR*	U *HDR*	U *HDR*	U *HDR*
TR					

Field Definitions

COMMAND

For information, see Primary Commands.

U (Function)

Column position to the left of the HEADER field where you can enter any non-blank character to list all the IDs with that HEADER.

Note: Only one header may be selected at a time.

Chapter 10: Editors and Commands

This chapter documents the TDF editing functions and valid TDF commands.

The CA Telon editors are:

- Panel editor - Operates in one of the following modes:
 - Full screen edit
 - Line edit
- Custom code editor
- Presentation store editor

CA Telon commands are in two categories:

- Primary commands, that can be entered in the COMMAND field of a TDF screen
- Line commands, which can be entered in line command fields that appear on some TDF screens

Panel Editor

The panel editor is a tool used to create, modify, browse, and purge panel image and panel definition data in the TDF.

Editing modes

The panel editor can be used in either of two modes:

- **Full screen edit mode**— Presents the full panel image. Invoke edit commands by pressing PF keys.
- **Line edit mode**— Presents the panel with the TDF COMMAND field at the top and line numbers at the left. Invoke edit commands by entering primary commands in the COMMAND field, line commands in the line command fields, or by pressing PF keys.

See n and Line Commands for information about edit commands.

Invoking the Edit Mode

Full screen edit

You can invoke full screen edit mode in any of these ways:

- For a new panel, on the Panel Definition menu, enter **CR** (create) in the FUNCTION field or **PI** (panel image) in the ITEM field
- For an existing panel, on the Panel Definition menu you can:
 - Enter **UP** (update) in the FUNCTION field or **PI** (panel image) in the ITEM field
 - Enter **LI** (list items) in the ITEM field and, on the resulting screen, enter **U** (update) as a line command for a listed item
- When in line edit mode, enter **LINE EDIT** or **LINE OUT** (primary commands)

Line edit mode

You can invoke line edit mode in any of these ways:

- On the Panel Definition menu, enter **CR** (create) or **UP** (update) in the FUNCTION field and **PD** (panel data) in the ITEM field, and on the resulting screen enter **LINE EDIT** (primary command)
- On the Panel Definition menu, enter **LI** (list items) in the FUNCTION field and **PD** (panel definition) in the ITEM field, and on the resulting screen enter **LINE EDIT** (primary command)
- When in full screen mode, enter **LINE EDIT** or **LINE OUT** (primary commands)

General Rules for Updating Fields

All panel fields consist of two components:

- **Attribute**— A one-byte field that precedes the painted image. CA Telon uses the attribute to store display information about the field.
- **Painted image**— The value you entered or the value that the panel editor generates.

Sets of rules

Panel editing follows a set of general field update rules and a set of specific rules depending on the type of field:

- Variable field
- Literal field
- Field with panel data
- Wrapped variable field
- Wrapped literal field
- Long literal field

General rules

- Fields cannot overlap.
- A field cannot start in line 1, column 1 of a panel.
- If a field starts in column 1, the attribute byte for the field is at the end of the previous line.
- You can associate panel data with a field, using the Update Panel Fields (Online) screen or the Update Panel Fields (Batch) screen.

Note: In the panel editor, panel lines that contain fields with panel data (PD) information have special highlighting, protection, and update rules associated with them. These rules attempt to minimize the unintentional loss of user-entered data during the editing session.

- Update rules are applied line by line and are invoked based on the entire contents of the line, not just the contents being displayed.

Rule violation

If an update rule is violated, CA Telon:

- Highlights the invalid line and places the cursor there.
- Issues the message REQUESTED UPDATES NOT ALLOWED.

You can correct the update to the line, or you can use the RESET line command to cancel the update and redisplay the line.

Note: If you enter the RESET primary command, CA Telon cancels all updates on the screen and redisplay the screen as it was before updates.

Transposing fields

To transpose fields on a line:

1. Use the FIELD SPLIT command to split the line such that there is a new line for each field contained on the original line.
2. Use the MOVE and OVER line commands to resequence the fields on the original line.

Rules for Variable Fields

You designate a variable field by making the first character the image character that represents the usage of the field (input, output, outin, or select). See Update Session Controls for more information about image characters.

If you specify a number following the image character, that number is the field length. After you press Enter, the variable field is redisplayed as a number of image characters equal to the specified length minus one (for the attribute byte), provided there are no other fields already defined in the range which the requested length spans. If the specified length would extend the field beyond the column size of the panel (that is, beyond the rightmost column of the panel), CA Telon truncates the length so that the field ends in the last column of the panel.

If you enter only the image character, CA Telon determines the length of the field by locating the next space that follows the last character of the variable. CA Telon converts all intervening characters to the image character.

Rules for Literal Fields

A literal is a field that does not begin with a variable character. A literal field ends in one of these ways:

- One or more spaces followed by a variable character (field)
- Three or more spaces until the next literal character
- A literal break character

Rules for Fields with Panel Data

A field with panel data is a field for which specifications exist in the panel definition (that is, a field whose definition consists of more than an attribute byte and a painted image).

In **full screen edit** mode:

- Lines containing fields with panel data are highlighted
- If a line contains more than one field, the entire line is protected
- You can use the PD command to unprotect the lines that are protected
- If the screen contains a single field, then it is unprotected and you can update it

In **line edit** mode:

- Same rules used for full screen edit mode apply.
- Lines containing fields with panel data also display special messages in the line number field instead of a number. Possible messages and their meanings are:
 - ==PD=>- Single field with panel data.
 - =PROT=>- A protected line with two or more fields displayed, at least one having panel data.
 - =PD==>- An unprotected line with multiple fields.
 - ++PD+>- A line containing fields with panel data that are not displayed.

Use the PD command to redisplay line numbers.

Note: On color terminals, the color of protected lines is different from PD lines. Non-PD lines are displayed in the normal color.

Single fields and multiple fields

Panel update rules differ depending on whether the screen contains a single field or multiple fields.

You can make single fields longer or shorter or move them to any position on the line. If the field is a variable, you can change it to another type of variable by altering the field's first character.

If two or more fields are displayed, you can perform any one of these operations to update all displayed variables:

- Shift fields to the right or left by changing the starting column positions of the fields.
- Lengthen or shorten the field by adding characters to or deleting them from the right side. All variable types and starting positions must remain the same. Any character can be used to lengthen a field since it is converted into the variable's paint character.
- Change variables to another type by changing the first character. All starting positions and lengths must remain the same.
- Add or delete variables.

Note: To do these operations, fields with panel data must be unprotected. Use the PD primary command if necessary.

Adding and erasing fields

You add new fields by painting a new variable. You can delete fields by erasing their painted image. You cannot perform other changes to the other displayed variable fields when adding or erasing.

Literal fields with panel data

You can add, change, or delete a literal field with panel data. If the field does not change position and length, the panel data remains the same. If these changes do occur, CA Telon assigns the panel data only to fields whose starting positions do not change.

Rules for Wrapped Variable Fields

A wrapped variable field is a variable field that spans more than one line. A wrapped variable field consists of three components:

- One-byte head
- Middle section
- One-byte tail

There are special update rules associated with each section of a wrapped variable field.

To define a wrapped variable, enter the image character that represents the use of the field (input, output, outin, or select) followed by a number that specifies the length of the field. Then enter **W** to end the wrapped variable. For example, a complete entry might be **+100W**.

CA Telon creates the field with the specified length, provided that the field does not overlap an existing field on the initial line.

Note: If the field does not span a line, CA Telon creates a regular field.

Edit mode considerations

In full screen edit mode, wrapped fields are treated as if they have panel data.

In line edit mode, displays messages in place of line numbers of unprotected fields. The possible messages and their meanings are:

- **==WF=>**— Wrapped variable field with no panel data
- **=WPD=>**— Wrapped variable field with panel data
- **++WF+>**— Undisplayed wrapped variable field with no panel data
- **+WPD+>**— Undisplayed wrapped variable field with panel data

Update rules

- CA Telon ignores updates to the middle of a wrapped variable field.
- You can update the head or tail when the wrapped variable field is the only field on the line.
- If there is more than one field on the line, the line is unconditionally protected regardless of the protection status specified with the PD command).
- If the head is deleted, the entire wrapped variable field is deleted, except the blank lines that the middle and tail use.
- You can change to another type of wrapped variable by changing the first character of the head.

- You can change the length of a wrapped variable field only from the tail. CA Telon ignores length changes to the head. If you change the starting position of the head, CA Telon moves the entire field (this causes the tail to move).
- If the tail is deleted, it decreases the length of the field by the tail's originally displayed length. If the tail no longer starts in column 1 of the screen, the same thing happens.
- You change the length of the field by adding or deleting characters from the right side of the tail.
- The maximum field size is 240 bytes. If the field's length exceeds the maximum, CA Telon automatically truncates the field and displays a warning message.
- If you increase the panel size (columns) and a wrapped variable field can now fit on one line, CA Telon automatically unwraps the field.

Rules for Wrapped Literal Fields

Wrapped literal fields are literal fields that span more than one line. The maximum length allowed for a wrapped literal field is 240 bytes. No special update rules are associated with updating wrapped literal fields.

Exceeding maximum field length

If you create a wrapped literal that exceeds the maximum 240 bytes, END and SAVE command processing:

- Issues a warning message.
- Breaks the field into two (or more) literals.
- Every 240th byte of the original literal is lost because it becomes the attribute byte for the next literal field created by CA Telon as part of this processing.

Wrapped literals with panel data

Since wrapped literals are physically contained on two or more lines, each segment of the wrapped literal can have different panel data associated with it. However, during END or SAVE command processing, CA Telon uses only the first segment with panel data. If this condition is detected before END or SAVE command processing is initiated, CA Telon issues a warning message.

Rules for Long Literal Fields

A long literal field is a literal field that contains more than three spaces. For example, this literal contains six spaces:

PROJECTED UNITS IN STOCK - 30 DAYS:

Long literal fields are created on the Update Panel Fields (Online) screen or the Update Panel Fields (Batch) screen.

In full screen edit mode, lines containing long literals are highlighted. If the line contains more than one field, the line is unconditionally protected; if the long literal is the only field on the line, updates are allowed.

In line edit mode, messages are displayed in the line number field. The possible messages and their meanings are:

- **=WPD=>**— Long literal field (displayed).
- **+WPD+>**— Long literal field (not displayed).

Update rules

By definition, long literals have panel data associated with them and invoke special update rules:

- Variables cannot be added to a line containing a long literal.
- By default, all literals entered on a line are assumed to be part of a long literal. To place more than one literal on a line, use MOVE or COPY with OVER to move or copy a literal from another line.

Wrapped long literals

CA Telon supports wrapped long literals. However, if you create a long literal on the Update Panel Fields (Online) screen or Update Panel Fields (Batch) screen that spans more than one line and is not wrapped, it is automatically broken into two or more literal fields. The original panel data (that is, the original field attributes) is replicated for each segment of the field or fields created as a result.

Custom Code Editor

Use the custom code editor to create, modify, browse, and purge custom code stored in the TDF. Custom code members can be one of these types:

- COBOL
- PL/I
- JCL

The type of custom code member that you edit determines the default beginning column displayed in the editing session:

- COBOL members begin in column 7
- PL/I members begin in column 2
- JCL members begin in column 1

The display column value appears at the upper right of the screen during an editing session.

Note: If you want to view the leftmost columns when editing a COBOL or PL/I custom code member, you can enter the RIGHT command on the Command Line or press the RIGHT PF key if one has been defined.

Converting to JCL

To convert a COBOL or a PL/I member into a JCL member, change the member's start column to 01 and enter two forward slashes (//) as the first two characters on the first line of the member. When you save or submit the member, it is recognized as a JCL member.

Editing multiple members

You can edit multiple custom code members in a single editing session, if all members are of the same type. In a multiple-member editing session, a member line for each member in the session is displayed immediately before the first data line of the member. Thus, member lines serve as headers for the members that you are editing. A member line is signified by the appearance of this message in the line number field:

=MBR=>

The line contains a full identification of the member, including its total number of data lines.

Note: You can enter certain line commands in the line number field of a member line to edit the entire member.

Size considerations

The custom code editor displays a maximum of 32 lines and 80 columns per screen regardless of the 3270 terminal model used, or less, if limited by the 3270 model type.

The maximum size for a custom code member is 9,999 lines. If you edit a member that exceeds this limit, the custom code editor cannot perform END and SAVE command processing.

Invoking the custom code editor

You can invoke the custom code editor in any of these ways:

- On the Online Program Definition menu or the Nonterminal Program Definition menu, specify create, update, or list as the function, and enter **CC** in the ITEM field. For create or update, also identify the member in the CUSTCODE field.
- On the List/Show Custom Code screen, select a member to edit. You can access this screen by selecting the custom code option on any of these screens:
 - Create/Update Screen Definition
 - Create/Update Driver Definition
 - Create/Update IMS/DC Report Definition
 - Create/Update Nonterminal Definition
 - Create/Update Batch Definition
 - Create/Update Stored Procedure

When you are in the custom code editor, the word EDIT is displayed in the upper left corner of the screen.

Presentation Store Editor

Use the presentation store editor to create, modify, browse, and purge presentation stores in the TDF.

Editing multiple members

You can edit multiple presentation stores in a single editing session. In this case, a member line for each member in the session is displayed immediately before the first data line of the member. Thus, member lines serve as headers for the members that you are editing. A member line is signified by the appearance of this message in the line number field:

=MBR=>

The line contains a full identification of the member, including its total number of data lines.

Note: You can enter certain line commands in the line number field of a member line to edit the entire member.

Invoking the presentation store editor

Invoke the presentation store editor in one of the following ways:

- On the Prototyping Facility menu, enter **VI** (view) in the FUNCTION field and **PS** (presentation store) in the ITEM field
- On the Prototyping Facility menu, enter **LI** (list) in the FUNCTION field (and **PS** (presentation store) in the ITEM field) to display the List Presentation Stores screen, where you can enter **U** as a line command for the listing of the presentation store to be edited
- On the List Panel Definitions screen, enter **.VPS** in the COMMAND field to view the active presentation store

When you are in the presentation store editor, the word EDIT is displayed at the top left of the screen.

Primary Commands

Submit primary commands to a CA Telon editor to control the editing session.

Enter primary commands in the COMMAND field of the screen. If a primary command has been assigned to a PF key, you can invoke the command by pressing the key. See Update Session Controls for more information about assigning commands to PF keys.

When you edit a panel in full screen edit mode, you can invoke a primary command only by pressing a PF key because the COMMAND field is not displayed in full screen edit mode.

Not all primary commands are valid in each editor. Restrictions on individual commands are noted in the documentation of those commands.

Syntax conventions

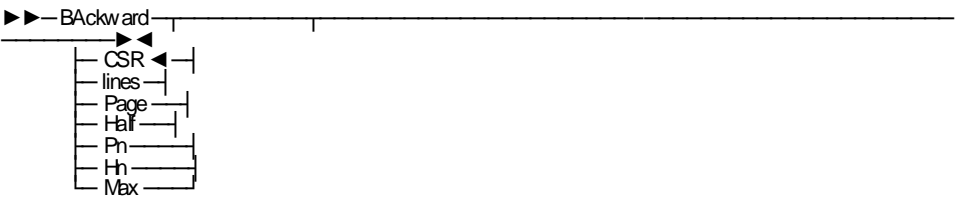
The following table contains the conventions for the primary command syntax in this section:

UPPERCASE ROMAN	Required keyword, or required part of a keyword
lowercase roman	Optional keyword, or optional part of a keyword
<i>lowercase italic</i>	Variable
{parameter}A {parameter}B {parameter}C	(<i>Required.</i>) One of these must be specified in the command.

[parameter]A	(<i>Optional.</i>) Valid but not required in the command.
[parameter]A [parameter]B [parameter]C	(<i>Optional.</i>) Any one is valid but none is required in the command.

BACKWARD

Command Format for BACKWARD



Syntax Definitions

Backward

Scroll backward the value displayed in the SCROLL field. (In the panel editor full screen mode, you can invoke BACKWARD only without options and only by pressing the assigned PF key.) You can modify the SCROLL field with one of these values:

CSR

(*Default.*) Scroll backward beginning at cursor position.

lines

Number of lines to scroll backward.

Page

Scroll backward one page.

Half

Scroll backward one half-page.

Pn

Number of pages to scroll backward.

Hn

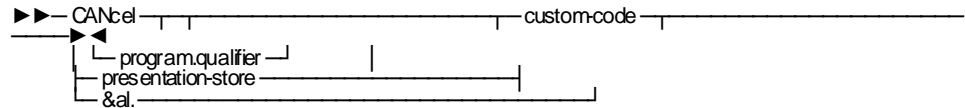
Number of half pages to scroll backward.

Max

Scroll backward to the beginning.

CANCEL

Command Format for CANCEL



Syntax Definitions

CANcel

End an editing session, deleting the edits made during the session, and return to the screen from which the session was requested.

program

Program header and identifier.

qualifier

To identify a program type (SD, ND, BD, RD, DR, SP).

custom-code

Removes the named custom code member from the editing session.

presentation-store

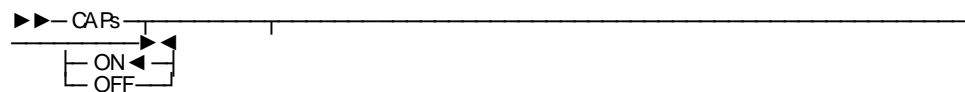
Removes the named presentation store from the editing session.

ALL

Cancels all custom code or presentation stores in an editing session.

CAPS

Command Format for CAPS



Syntax Definitions

CAPs

Specifies the case of characters you input during the editing session.

ON

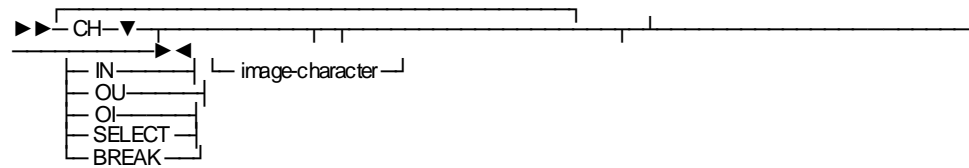
(Default.) All characters are entered in uppercase.

OFF

Characters can be entered in mixed lower and uppercase.

CH

Command Format for CH



Syntax Definitions

CH

(Panel editor line edit mode only.) Changes a character that identifies the usage of a panel field or changes the literal break character. You can change more than one character in a single command; for example, entering **CH IN ? OUT X** changes the image characters for input and output fields to ? and X, respectively.

IN

Field used for input only.

OU

Field used for output only.

OI

Field used for either output or input.

SELECT

Field used for selection by the application user.

BREAK

Literal break character.

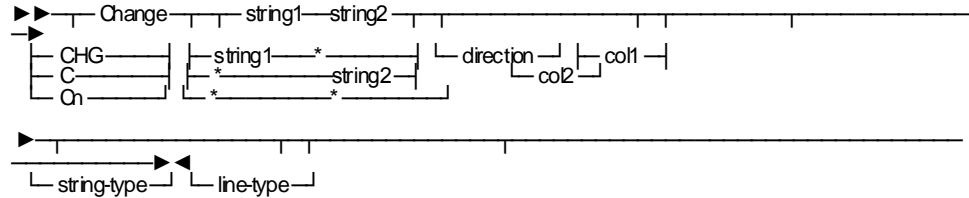
image-character

Character to represent the specified usage. Characters that you *cannot* use as image characters are:

- (— Left parenthesis
-)— Right parenthesis
- ,— Comma
- '— Single quotation mark

CHANGE

Command Format for CHANGE



Syntax Definitions

Change |CHG|C|Cn

Changes a specified character string to another specified character string. (Custom code members and presentation stores only.)

Note: *Cn* saves the CHANGE command you are entering. The value of *n* can be 1 to 9; therefore, you can save 10 CHANGE commands (including C itself) for the editing session. You can invoke a saved CHANGE command with the REPEAT CHANGE command.

string1

String to be changed.

string2

String to replace *string1*.

*

String specified for this parameter in the most recent prior CHANGE command issued during this session.

direction

Controls the direction and starting point in which to perform the change operation. Values are:

- ALL— Search begins at the TOP-OF-DATA line and proceeds forward to find all matches with *string1* until BOTTOM-OF-DATA line is reached.
- FIRST— Search begins at the TOP-OF-DATA line and proceeds forward to find the first *string1* match.
- LAST— Search begins at the TOP-OF-DATA line and proceeds forward to find the last *string1* match.
- NEXT— (*Default.*) Search begins at the current cursor location and proceeds forward to find the next *string1* match.
- PREV— Search begins at the current cursor location and proceeds backward until the *string1* match occurs.

col1 col2

Range of columns searched for occurrences of *string1*. *Col1* delimits the left of the range; *col2* delimits the right of the range.

string-type

Type of string on which to perform the change operation. Values are:

CHARS

(Default.) Considered successful regardless of what precedes or follows *string1*.

PREFIX

Considered successful only if *string1* is not preceded with a word-character and is followed with a word-character.

SUFFIX

Considered successful only if *string1* is preceded with a word-character and is not followed with a word-character.

WORD

Considered successful only if *string1* is not preceded and not followed with a word-character.

line-type

Type of line on which to perform the change operation. Values are:

ALL

(Default.) All lines.

X

Excluded lines only.

NX

Non-excluded lines only.

In the Data Group, the CHANGE and REPEAT CHANGE commands must be used with extreme care.

Telco CANNOT change: Inherited key values (marked by "@"), Table labels, Database labels, File labels.

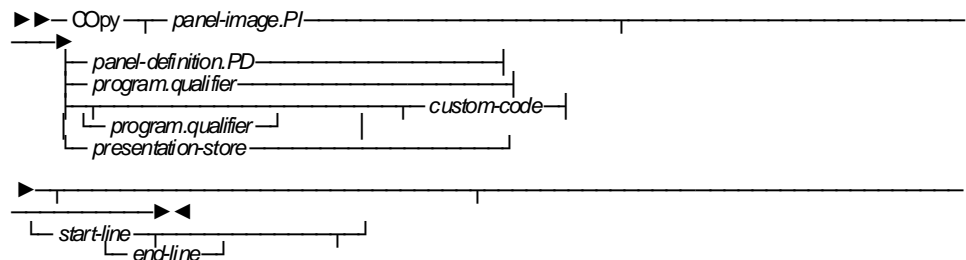
Telco CAN change: Non-inherited key values, Segment labels, Row labels, Record labels.

When a change or repeat-change command is issued, the unchangeable items may APPEAR to change, but when the screen is saved, they revert to their original values. In the following example, the command CHANGE ALL EMPL XXX was issued, then the Data Group was saved and updated again. Note the occurrences of "EMPL" that remain, and the occurrences of "XXXX" when the changes took place. Note also the values in the IGNORE column that tie the rows, records, segments, and so on, to their respective tables, files and databases (PCB's).

	LABEL	REQUEST	KEY/WHERE	IGNORE
TAB=>	TRGEMPL		TELON. TRGEMPL	
	%%%%%%%%	%%%%%%%%	%%%%%%%%	%%%%%%%%
ROW=>	TRGEMPLB	@DUMMY	@EMPL - ID	TRGEMPL
VSAM=>	TRGEMPLB			
	%%%%%%%%	%%%%%%%%	%%%%%%%%	%%%%%%%%
REC=>	TRGXXXB	@DUMMY	XFER-XXXX- ID	TRGEMPLB
PCB=>	TRGEMPL	TRGXXX		
	%%%%%%%%	%%%%%%%%	%%%%%%%%	%%%%%%%%
SEG=>	TRGXXX	@DEFINE	@XFER- EMPL - ID	TRGEMPL

COPY

Command Format for COPY



Syntax Definitions

COPY

Copies text from the named entity into the entity that you are editing. Also, issue either the AFTER line command or the BEFORE line command to specify where the copied lines are placed.

For information about line commands, see Line Commands.

panel image

Panel image header and identifier.

panel-definition

Panel definition header and identifier.

program

Program header and identifier.

qualifier

To identify a program type (SD, ND, BD, RD, DR, SP).

custom-code

Custom code member name. (When a COPY command is issued for a copy member alone, it is copied from the same program as the copy member the command is being issued from.)

presentation-store

Presentation store name.

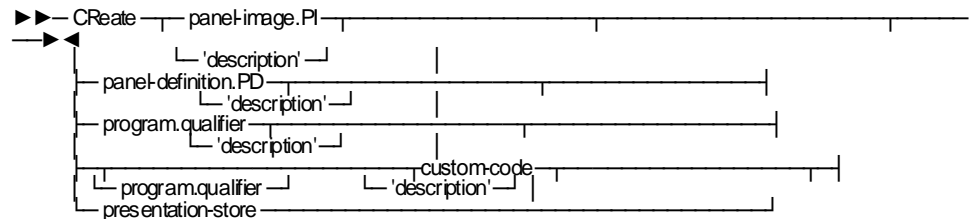
start-line

First line to be copied.

end-line

Last line to be copied.

CREATE

Command Format for CREATE

Syntax Definitions

CReate

Create the entity you are editing by copying the named entity. Also, use either the COPY or the MOVE line command to specify the lines that you want to use for the create.

panel-image

Panel image header and identifier.

panel-definition

Panel definition header and identifier.

program

Program header and identifier.

qualifier

To identify a program type (SD, ND, BD, RD, DR, SP).

custom-code

Custom code member name.

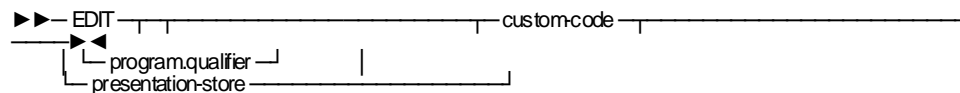
Description

Panel or custom code member description.

presentation-store

Presentation store name.

EDIT

Command Format for EDIT

Syntax Definitions

EDIT

Performs one of these functions:

- In show mode, transfers to edit mode, if entity is named
- Adds the named entity to the current editing session

program

Program header and identifier.

qualifier

To identify a program type (SD, ND, BD, RD, DR, SP).

custom-code

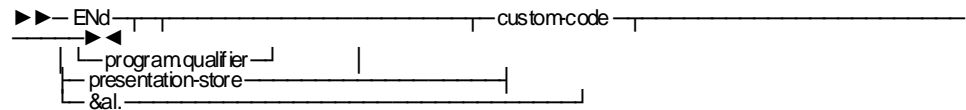
Custom code member to edit. If the member does not exist, a new member is created for the current screen definition.

presentation-store

Presentation store to edit. If it does not exist, a blank presentation store with the name you specified is created for you to edit.

END

Command Format for END



Syntax Definitions

ENd

Ends the current editing session, saves the edits you made during the session, and returns you to the screen where you requested the session.

program

Program header and identifier.

qualifier

To identify a program type (SD, ND, BD, RD, DR, SP).

custom-code

Ends editing of the custom code member, but if you are editing multiple members, does not end the editing session.

Note: To end an editing session successfully, no member that you are editing can exceed 9,999 lines.

presentation-store

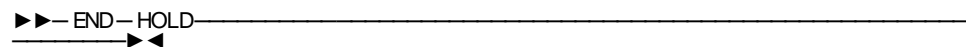
Ends editing of the presentation store, but if you are editing multiple members, does not end the editing session.

ALL

Ends an editing session in which you are editing multiple members.

END HOLD

Command Format for END HOLD



Syntax Definitions

END HOLD

Ends the current TDF session and transfers to the previous held session.

EQUATE

Command Format for EQUATE

►►—Equate—location-name—
→►◄

Syntax Definitions

Equate

Equates a name with a line number in the entity being edited. (Not valid in the full screen mode or the panel editor.)

location-name

One- to eight-character name assigned to the first numbered line currently at the top of the screen (or, if it is displayed for a single edit, to the TOP OF DATA line, which is line number 000000).

The following actions cause the location name to be deleted:

- Deleting a member where *custom-code* matches *location-name*
- Deleting a line to which *location-name* has been assigned

FIND

Command Format for FIND

►►—Find—string—
┌ F ─┘ ┌ direction ─┘ ┌ string-type ─┘ ┌ line-type ─┘
└ Fn ─┘
►—
┌ col1 ─ col2 ─┘

Syntax Definitions

Find|F|Fn

(Custom code members and presentation stores only.) Finds the specified character string.

Note: Saves the FIND command you are entering. The value of *n* can be 1 to 9; therefore, you can save 10 FIND commands (including F itself) for the editing session. You can invoke a saved FIND command with the REPEAT FIND command.

string

String to be located. Search string rules are the same as for the ISPF editor except that a leading left parenthesis must be delimited by single quotes, as shown here:

'('

direction

Controls the direction and starting point in which to perform the find operation. Values are:

- ALL — Search begins at the TOP-OF-DATA line and proceeds forward to find all matches with *string1* until BOTTOM-OF-DATA line is reached.
- FIRST— Search begins at the TOP-OF-DATA line and proceeds forward to find the first *string1* match.
- LAST— Search begins at the TOP-OF-DATA line and proceeds forward to find the last *string1* match.
- NEXT— (*Default.*) Search begins at the current cursor location and proceeds forward to find the next *string1* match.
- PREV— Search begins at the current cursor location and proceeds backward until the *string1* match occurs.

string-type

The type of string on which to perform the find operation. Values are:

- CHARS
- PREFIX
- SUFFIX
- WORD

line-type

Type of line on which to perform the find operation. Values are:

ALL

(*Default.*) All lines.

X

Excluded lines only.

NX

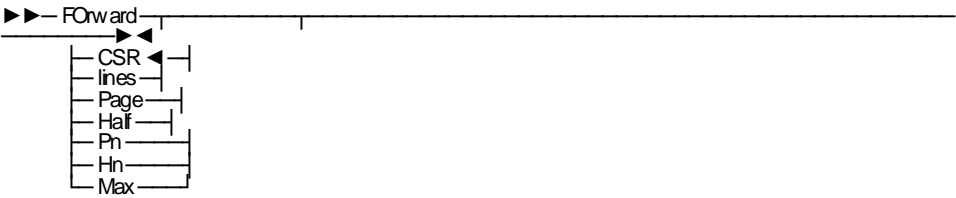
Nonexcluded lines only.

col1 col2

Range of columns to be searched for occurrences of *string*. *Col1* delimits the left of the range; *col2* delimits the right of the range.

FORWARD

Command Format for FORWARD



Syntax Definitions

Forward

Scroll forward the value displayed in the SCROLL field. (In the panel editor full screen mode, you can invoke FORWARD only without options and only by pressing the assigned PF key.) You can modify the SCROLL field with one of these values:

CSR

(Default.) Scroll forward beginning at cursor position.

lines

Number of lines to scroll forward.

Page

Scroll forward one page.

Half

Scroll forward one half-page.

Pn

Number of pages to scroll forward.

Hn

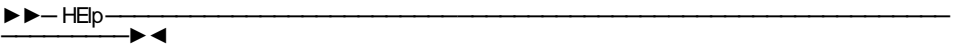
Number of half pages to scroll forward.

Max

Scroll forward to the end.

HELP

Command Format for HELP



Syntax Definitions

HElp

Displays help information about the current screen.

HOLD

Command Format for HOLD

►►—HOLD—
————►◄

Syntax Definitions

HOLD

Stores the current screen in its current state and transfers control to the TDF Main menu. You can restore the screen with the SWAP command.

INSTALL

Command Format for INSTALL

►►—INSTALL—
————►◄

Syntax Definitions

INSTALL

Transfers control to the TDF Installation screen.

ISPF/PDF

Command Format for ISPF/PDF

►►—ISPF/PDF—command—
————►◄

Syntax Definitions

ISPF/PDF

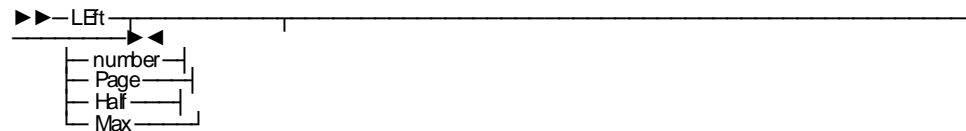
(Not valid in PWS.) Submits a specified command to ISPF/PDF. If no such command is specified, transfers control to TDF Main menu.

command

Command to be executed.

LEFT

Command Format for LEFT



Syntax Definitions

LEft

Specifies the leftmost column of the initial screen display, thereby shifting the screen display to the right. If no option is specified, the default beginning column number of the display is:

- 02 for a panel
- 07 for a COBOL custom code member
- 02 for a PL/I custom code member
- 01 for a JCL custom code member

This is not a valid command for editing a presentation store.

number

Shift the display right by the number of columns specified, not to exceed position 1.

Page

Shift the display right by the number of columns equal to the full size of the terminal screen or until column 1 appears.

Half

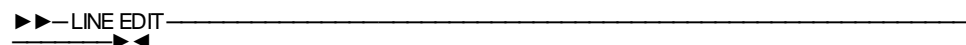
Shift the display right by the number of columns equal to half the size of the terminal screen.

Max

Shift the display right until column 1 appears.

LINE EDIT

Command Format for LINE EDIT



Syntax Definitions

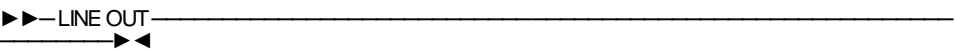
LINE EDIT

(Panel editor only.) Switches the editing session between line edit mode and full screen edit mode. Changes made in the session are saved when you switch modes.

Issued from the Update Panel Fields (Online) screen and the Update Panel Fields (Batch) screen, this command transfers you to line edit mode.

LINE OUT

Command Format for LINE OUT



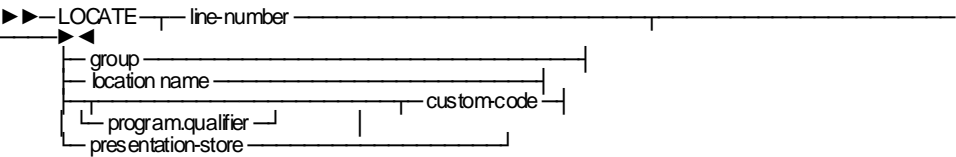
Syntax Definitions

LINE OUT

(Panel editor only.) Performs the same function as LINE EDIT.

LOCATE

Command Format for LOCATE



Syntax Definitions

Locate

Redisplays the screen at the specified location.

line-number

Number of the line.

custom code

Name of the custom code member.

group

(Panel editor only.) Beginning of the named batch group.

location-name

Location previously named in an EQUATE command.

program

Program header and identifier.

qualifier

To identify a program type (SD, ND, BD, RD, DR, SP).

custom-code

Beginning of the named custom code member.

presentation-store

Beginning of the named presentation store.

MBCOPY (MARK AND BOUND COPY)

Command Format for MARK AND BOUND COPY

►►—MBCopy—first-line—first-column—last-line—last-column—►
 ►—target-line—target-column—►►◀

Syntax Definitions

MBCopy

(Panel editor line mode and custom code members only.) Copies a marked and bounded block of fields from one location in the entity you are editing to another but only if the copy will not cause existing fields to be overlaid.

For example, if you enter **MBC 7 2 9 40 15 2**, fields contained in the block in the line range 7-9 and column range 2-40 are copied to the same relative positions beginning at line 15, column 2, if the copy action will not cause existing fields to be overlaid.

first-line

Line where the block begins.

first-column

Leftmost column in *first-line* where the block begins.

last-line

Line where the block ends.

last-column

Rightmost column in *last-line* where the block ends.

target-line

Line on which the copy is marked and block is to begin.

target-column

Leftmost column on *target-line* where the copy is marked and block is to begin.

MBMOVE (MARK AND BOUND MOVE)**Command Format for MBMOVE (MARK AND BOUND MOVE)**

►►—MBMove—first-line—first-column—last-line—last-column—►
 ►—target-line—target-column—►►

Syntax Definitions**MBMove**

(Panel editor line mode and custom code members only.) Moves a marked and bounded block of fields from one location in the entity you are editing to another but only if the move will not cause existing fields to be overlaid.

For example, if you enter **MBM 7 2 9 40 15 2**, fields contained in the block in the line range 7-9 and column range 2-40 are moved to the same relative positions beginning at line 15, column 2, if the move action will not cause existing fields to be overlaid.

first-line

Line where the block begins.

first-column

Leftmost column in *first-line* where the block begins.

last-line

Line where the block ends.

last-column

Rightmost column in *last-line* where the block ends.

target-line

Line on which the copy is marked and block is to begin.

target-column

Leftmost column on *target-line* where the copy is marked and block is to begin.

MBPURGE (MARK AND BOUND PURGE)

Command Format for MBPurge (MARK AND BOUND PURGE)

►►— MBPurge —first-line—first-column—last-line—last-column —►◄

Syntax Definitions

MBPurge

(Panel editor line mode and custom code members only.) Deletes a marked and bounded block of fields from the specified location in the entity you are editing.

For example, if you enter **MBP 7 2 9 40**, fields contained in the block in the line range 7-9 and column range 2-40 are deleted.

first-line

Line where the block begins.

first-column

Leftmost column in *first-line* where the block begins.

last-line

Line where the block ends.

last-column

Rightmost column in *last-line* where the block ends.

MENU

Command Format for MENU

►►— MEnu —————►◄

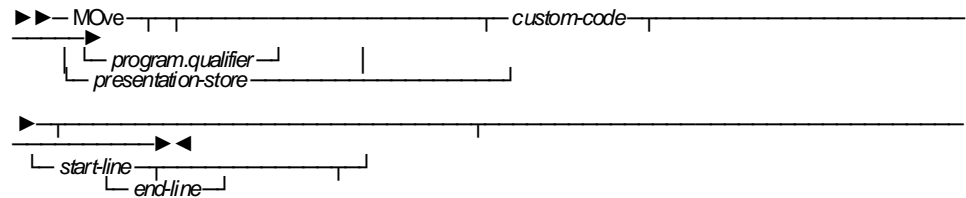
Syntax Definitions

MEnu

Ends the current editing session, saves the edits, and transfers control to the TDF Main menu.

MOVE

Command Format for MOVE



Syntax Definitions

MOVe

Moves text from the named entity into the entity that you are editing. Also, issue either the AFTER line command or the BEFORE line command to specify where the moved lines are to be placed.

program

Program header and identifier.

qualifier

To identify a program type (SD, ND, BD, RD, DR, SP).

custom-code

Name of the custom code member.

presentation-store

Name of the presentation store.

start-line

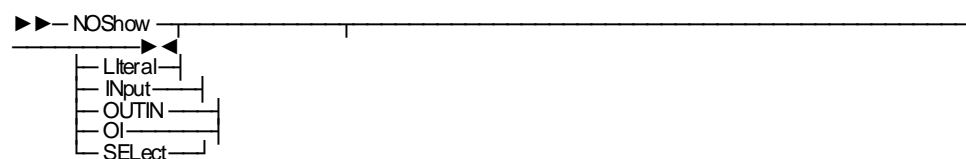
First line to be moved.

end-line

Last line to be moved.

NOSHOW

Command Format for NOSHOW



Syntax Definitions

NOShow

(Panel definition edit mode only). Excludes the requested field types from the display.

Literal

Excludes literal fields from the display.

INput

Excludes input fields from the display.

OUTput

Excludes output fields from the display.

OUTIN

Excludes outin fields from the display.

OI

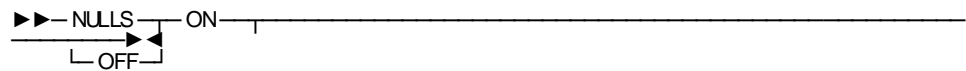
(Same as OUTIN). Excludes outin fields from the display.

SElect

Excludes select fields from the display.

NULLS

Command Format for NULLS



Syntax Definitions

NULLs

Specifies whether trailing spaces are nulls or blanks. Entered without an option, this command toggles between the two specifications.

ON

Trailing spaces are null. You can enter characters in these spaces without first pressing EOF.

OFF

Trailing spaces are blanks. You must erase these blanks by pressing EOF before you can enter characters in these spaces.

PANEL COLUMNS

Command Format for PANEL COLUMNS

▶▶—PANELC—columns—
→▶◀

Syntax Definitions

PANELC *columns*

(Panel editor line edit mode only.) Changes the width of the panel you are editing to *columns*. However, if a loss of data would result, CA Telon displays an error message and no change occurs.

PANEL LINES

Command Format for PANEL LINES

▶▶—PANELLLine—lines—
→▶◀

Syntax Definitions

PANELLLine *lines*

(Panel editor line edit mode only.) Changes the depth of the panel you are editing to *lines*.

When using a Model 3 or Model 4 terminal, if you modify a panel to 24 lines, the next time you update this panel it is displayed in a Model 2 mode. If you ended your current editing session and began a new editing session, this change of display mode would still be present. You can return to your original display mode in one of these ways:

- SWAP EDIT
- UPDATE GROUP
- Change panel lines to any number other than 24, end the editing session, and begin a new editing session

PANEL SIZE

Command Format for PANEL SIZE

▶▶—PANELSize—lines—columns—
→▶◀

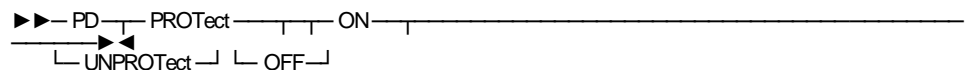
Syntax Definitions

PANELSize *lines columns*

(Panel editor line edit mode only.) This command allows you to perform the functions the PANEL LINES and PANEL COLUMNS commands in a single command.

PD

Command Format for PD



Syntax Definitions

PD

(Panel editor only.) Protects or unprotects fields defined on the Update Panel Fields (Online) screen or the Update Panel Fields (Batch) screen. In full screen edit mode, the PF key to which PD is assigned acts as a toggle switch between PD PROTECT and PD UNPROTECT. In the line edit mode, the default values are PROTECT ON.

A line is protected if two or more fields are displayed and one of those fields:

- Has panel data
- Is a wrapped variable
- Is a long literal

If PROTECT is in effect, the line number is replaced with:

=PROT>

If UNPROTECT is in effect, the line number is replaced with:

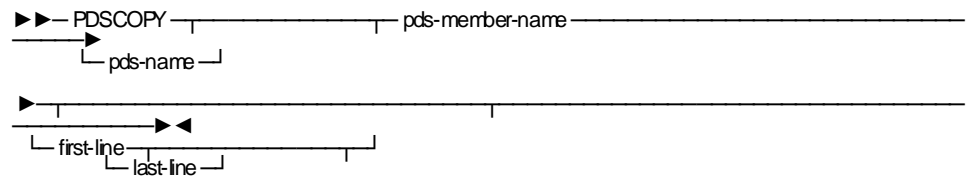
- ==WF=>— If the line contains a wrapped variable without panel data
- =WPD=>— If the line contains a wrapped variable or long literal
- ==PD=>— If the line contains a single field with panel data (these lines are automatically unprotected) or any other data
- +++PD+>— If the line contains one or more fields with panel data that are not displayed
- ++WF+>— If there is an undisplayed wrapped variable without panel data
- +WPD+>— If the line contains a wrapped variable with panel data or a long literal that is not displayed

PDF

See ISPF/PDF.

PDSCOPY

Command Format for PDSCOPY



Syntax Definitions

PDScopy

(Custom code editor only; not valid for PWS.) Copies all or part of a specified member in a partitioned data set into the member that you are currently editing, at a location you must specify with either the BEFORE or AFTER line command.

pds-name

Name of the partitioned data set. If not specified, the default name is the value in the PDSCOPY DSNAME field on the Update Session Controls screen.

pds-member-name

Name of the member to be copied from the partitioned data set.

first-line

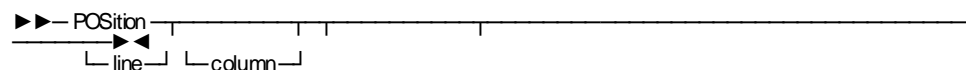
Line number of the first line to be copied from the member of the partitioned data set. If not specified, the default is line 1.

last-line

Line number of the last line to be copied from the member of the partitioned data set. If not specified, the default is the last line of the member.

POSITION

Command Format for POSITION



Syntax Definitions

POSiTion

(Panel definition edit mode only.) Redisplays the panel field rows beginning with the field that matches or is nearest to the line and column specified.

line

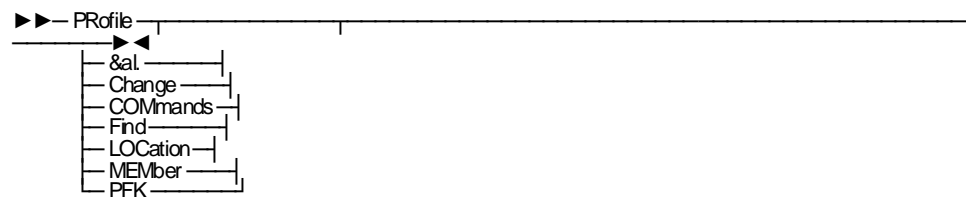
Line number of the field with which the list begins.

column

Column number of the field on the line value specified.

PROFILE

Command Format for PROFILE



Syntax Definitions

PRofile

Displays the profile of the current editing session. If you do not qualify the command, the profile information is:

- For editing panels:
 - Caps
 - Nulls
 - PD protect/unprotect
 - PD on/off
 - Variable characters
 - Literal break character
- For editing custom code members and presentation stores:
 - Caps
 - Nulls
 - Members

ALL

Lists all the information available for the editing session. For panel editing in full screen mode, use the assigned PF key to enter PROFILE ALL.

Change

List CHANGE commands entered during this editing session.

COMmands

Lists the incomplete line commands and line commands currently awaiting execution.

Find

List FIND commands entered during this editing session.

GROUP

(Panel editing.) Lists information on the batch groups that have been defined.

LOCATION

Lists the current location names that have been defined with the EQUATE command.

MEMber

Names of the members you are editing.

PFK

Displays the current PF key definitions.

PURGE

Command Format for PURGE

►►—Purge—member-name—
————►◄

Syntax Definitions

PURge

Confirms a purge that you have requested by:

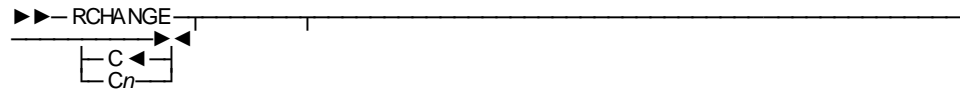
- Entering **PU** in the FUNCTION field of the Panel Definition menu.
- Entering **PU** in the FUNCTION field of the Online Program Definition menu.
- Entering **P** as a line command for a member listing on the List Presentation Stores screen.

member-name

Name of the custom code member or presentation store to be purged.

RCHANGE (REPEAT CHANGE)

Command Format for REPEAT CHANGE



Syntax Definitions

RCHANGE

Repeats a specified CHANGE command. Alone, it executes the most recent CHANGE command.

C

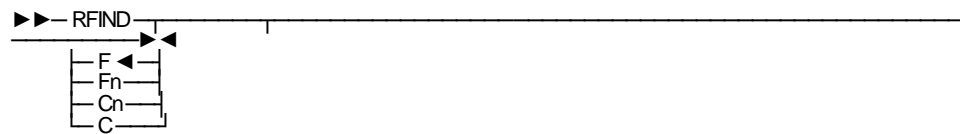
(Default.) The most recently executed CHANGE command.

Cn

A CHANGE command that you saved during a previous execution.

RFIND (REPEAT FIND)

Command Format for REPEAT FIND



Syntax Definitions

RFIND

Repeats a specified FIND command or the FIND portion of a CHANGE command. Alone, it executes the most recent FIND command.

F

(Default.) The most recently executed FIND command.

Fn

A FIND command that you saved during a previous execution.

C

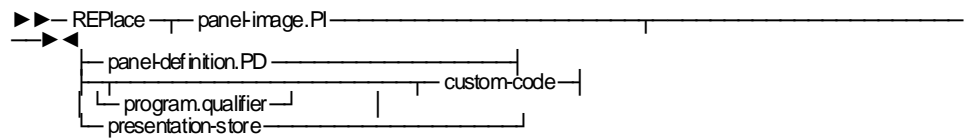
FIND portion of the most recently executed CHANGE command.

Cn

FIND portion of a CHANGE command that you save during a previous execution.

REPLACE

Command Format for REPLACE



Syntax Definitions

REPlace

Replaces the named entity with part or all of the entity that you are editing. Use the COPY or MOVE line command to specify the part of the entity you are editing that is to replace the named entity.

See Line Commands for information about line commands.

panel image

Panel image header and identifier.

panel-definition

Panel definition header and identifier.

program

Program header and identifier.

qualifier

To identify a program type (SD, ND, BD, RD, DR, SP).

custom-code

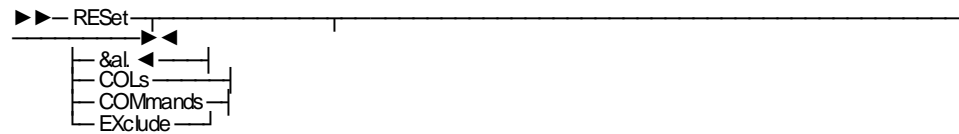
Name of the custom code member.

presentation-store

Name of the presentation store.

RESET

Command Format for RESET



Syntax Definitions

RESet

Cancels the specified group of line commands.

ALL

(Default.) Causes all COLS, line commands, EXCLUDE commands to become inactive.

COLS

All COLS commands.

COMmands

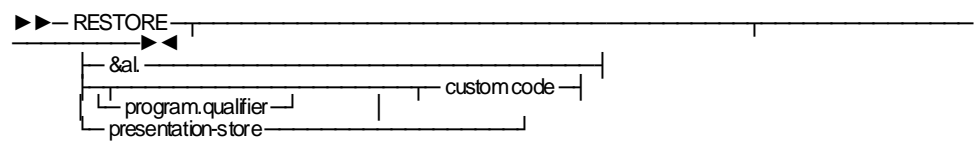
All pending line commands or line commands currently in force.

EXclude

All EXCLUDE commands.

RESTORE

Command Format for RESTORE



Syntax Definitions

RESTORE

For the current entity, cancels all edits that you made since you entered the current editing session or since the last SAVE command, and restores the current editing session to its state when you entered it.

ALL

For editing multiple custom code members or presentation stores, performs restore processing on all members in the session.

program

Program header and identifier.

qualifier

To identify a program type (SD, ND, BD, RD, DR, SP).

custom-code

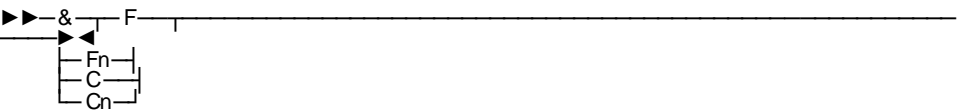
Custom code member to be restored.

presentation-store

Presentation store to be restored.

RESTORE FIND

Command Format for RESTORE FIND



Syntax Definitions

&

Redisplays a FIND command or a CHANGE command.

F

Last FIND command that was entered in this session.

Fn

FIND command defined when previously entered.

C

Last CHANGE command that was entered in this session.

Cn

CHANGE command defined when previously entered.

RESUME

Command Format for RESUME

▶▶—RESUME▶◀

Syntax Definitions

RESUME

Ends the current TDF session and transfers to the previous held session.

RIGHT

Command Format for RIGHT

▶▶—Right▶◀
|—number—|
|—Page—|
|—Half—|
|—Max—|

Syntax Definitions

RIght

Shift display left but never further than the rightmost position of the custom code member or panel image. If no option is specified, the default beginning column number of the display is:

- 02 for a panel
- 07 for a COBOL custom code member
- 02 for a PL/I custom code member
- 01 for a JCL custom code member

This is not a valid command for editing a presentation store.

number

Shift the display left by the number of columns specified, but not further than the rightmost position of the custom code member or panel image.

Page

Shift display left by the number of columns equal to the size of the terminal screen or until column 1 appears.

Half

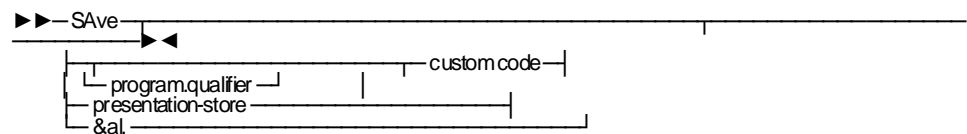
Shift the display left by the number of columns equal to half the size of the terminal screen.

Max

Shift the display left until the rightmost position of the member or panel image appears.

SAVE

Command Format for SAVE



Syntax Definitions

SAve

Saves the edits you have made during the session.

program

Program header and identifier.

qualifier

To identify a program type (SD, ND, BD, RD, DR, SP).

custom-code

Custom code member whose edits are to be saved.

Note: To be saved successfully, the custom code member cannot exceed 9,999 lines. We strongly recommend that no custom code member exceed 2,000 lines.

presentation-store

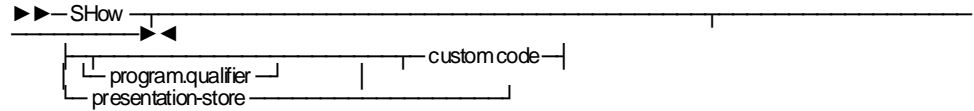
Presentation store whose edits are to be saved.

ALL

Saves edits to all members of a multiple-member editing session.

SHOW

Command Format for SHOW



Syntax Definitions

SHoW

If you are in edit mode, switches you to browse mode.

program

Program header and identifier.

qualifier

To identify a program type (SD, ND, BD, RD, DR, SP).

custom-code

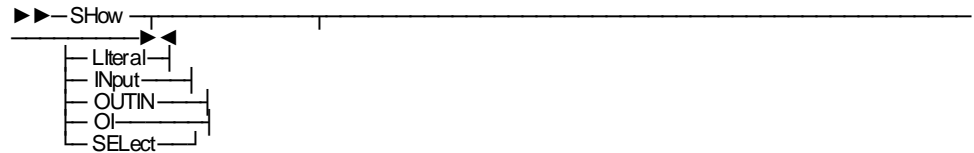
Adds a custom code member to the editing session.

presentation-store

Adds a presentation store to the editing session.

SHOW— Panel Definition Edit Mode

Command Format for SHOW



Syntax Definitions

SHoW

(Panel definition edit mode only). Displays the requested field types.

LIteral

Displays requested literal fields.

INput

Displays requested input fields.

OUTput

Displays requested output fields.

OI

(Same as OUTIN). Displays requested outin fields.

SElect

Displays requested select fields.

SUBMIT

Command Format for SUBMIT

▶▶—SUBmit—
————▶◀◀

Syntax Definitions

SUBmit

(Custom code only.) Submits the member that you are currently editing for execution. The member name must be of JCL type and must be the only member being edited in the current editing session.

To convert a COBOL or a PL/I member into a JCL member, change the member's start column to column 01 and enter two forward slashes (//) as the first two characters on the first line of the member.

SWAP

Command Format for SWap

▶▶—SWap—
————▶◀◀

Syntax Definitions

SWap

Transfers from the current session to the previously-held session.

Note: Press Enter to reactivate the held session.

SWAP EDIT

Command Format for SWAP

▶▶—SWAP—EDIT—
————▶◀◀

Syntax Definitions

SWAP EDIT

(Panel editor only.) Transfers to the Update Panel Fields (Online) screen.

Note: *From* the Update Panel Fields (Online) screen, this command switches the session to full screen edit mode.

TRANSFER

Command Format for TRANSFER

►►=n_____►◄

Syntax Definitions

=*n*

Transfers to another TDF screen. Valid *n* values are:

1

User Profile Maintenance menu

1C

Color Profile Maintenance Menu (PWS only)

1D

Update Program Definition Defaults

1P

Update Environment Definition Defaults

1S

Update Session Controls

2

Data Administration menu

3

Panel Definition menu

4

Online Processing menu

4N

Nonterminal Program Definition menu

4S

Online Program Definition menu

- 5**
Batch Program Definition menu
- 6**
Prototyping Facility menu
- U**
Utilities menu
- X**
(Exit the TDF)

TSO

Command Format for TSO

▶▶—TSO—tso-command—
————▶◀

Syntax Definitions

TSO *tso-command*

Submits the specified command to TSO for execution.

Note: This command is not valid for PWS or when running under CICS.

Line Commands

Submit line commands to a CA Telon editor to control the editing session.

You enter line commands in the line command field of the screen. (Not all screens have line command fields.)

Not all line commands are valid in each editor. Restrictions on individual commands are noted in the documentation of those commands.

PF key assignments

If a line command has been assigned to a PF key, you can invoke the command by pressing the key. PF key assignments are made on the Update PF Keys Definition screen. This is the correspondence between the short form of the line command listed on the screen and the full name of the command as presented in this section:

- **D**— DELETE
- **I**— INSERT
- **R**— REPEAT
- **C**— COPY
- **M**— MOVE
- **A**— AFTER
- **B**— BEFORE
- **O**— OVER
- **)**— RIGHT SHIFT
- **(**— LEFT SHIFT
- **X**— EXCLUDE
- **XX**— EXCLUDE (block)
- **F**— FIRST
- **L**— LAST
- **COLS**— COLS
- **FS**— FIELD SPLIT
- **LC**— LINE CLEAR
- **D**— DEFINE GROUP
- **U**— UPDATE GROUP
- **DG**— DELETE GROUP

See Update PF Keys Definition for information about assigning commands to PF keys.

When you edit a panel in full screen edit mode, you can invoke a line command only by pressing a PF key because no line command field is displayed in full screen edit mode.

Syntax conventions

These are the conventions for the primary command syntax in this section:

UPPERCASE ROMAN	Required keyword
<i>lowercase italic</i>	Variable
[parameter]A	An optional parameter; it is valid but not required in the command
	A separator to indicate a choice between the command to the left of the separator and a command to the right
ZZ . . . ZZ	Where ZZ is the command, this indicates that ZZ is entered on two lines and the operation it invokes is performed on the range of lines thus marked

AFTER

Syntax: **A** $[n]$

Function: Specifies the location after which the text is to be moved or copied, in conjunction with the MOVE or COPY command. N specifies the number of occurrences of marked text to move or copy after this location. Values for n are 1 to 9. If n not specified, the default is 1.

AFTER MULTIPLE

Syntax: **AM** $[n]$

Function: Specifies one or more locations after which lines are to be copied or moved, in conjunction with the COPY or MOVE line command.

If you enter **AM** or **AM n** on only one line, this command functions exactly as the AFTER line command does.

Unlike the AFTER command, AFTER MULTIPLE command allows you to specify, at the same time, multiple locations for a line or block of lines to be copied or moved. If you enter **AM** or **AM n** on multiple lines, the line or block of lines marked for copy or move is copied or moved after each line at which you enter a form of the AFTER MULTIPLE command. The line or block of lines will be repeated n times following a line at which you enter **AM n** .

BEFORE**Syntax:** **B**[*n*]

Function: Specifies the location before which the text is to be moved or copied, in conjunction with the MOVE or COPY command. *N* specifies the number of occurrences of marked text to move or copy before this location. Values for *n* are 1 to 9. If *n* not specified, the default is 1.

BEFORE MULTIPLE**Syntax:** **BM**[*n*]

Function: Specifies one or more locations before which lines are to be copied or moved, in conjunction with the COPY or MOVE line command.

If you enter **BM** or **BMn** on only one line, this command functions exactly as the BEFORE line command does.

Unlike the BEFORE command, BEFORE MULTIPLE allows you to specify, at the same time, multiple locations for a line or block of lines to be copied or moved. If you enter **BM** or **BMn** on multiple lines, the line or block of lines marked for copy or move is copied or moved before each line at which you enter a form of the BEFORE MULTIPLE command. The line or block of lines will be repeated *n* times before a line at which you enter **BMn**.

BOX**Syntax:** **BOX**[*n*] | **BOX***[*n*] | **BOX-**[*n*]

Function: (Custom code editor only.) Inserts a formatted box into the member that you are currently editing, allowing you to enter comments into the boxed area.

In COBOL members:

- Entering **BOX** without a qualifier creates a box, bounded by asterisks, that contains three blank lines into which you can insert comments. Use *n* to override the default three lines.
- Entering **BOX*n** creates a box, bounded by asterisks in columns 7 and 70 that contains *n* lines.
- Entering **BOX-n** creates a box, bounded by hyphens in that contains *n* lines. An asterisk at the beginning and end of the line identifies each comment line.

In PL/I members, the PL/I comment characters /* and */ delimit the boundaries of the box.

In JCL, the JCL comment characters (/*) delimit the boundaries of the box.

CANCEL**Syntax:** **CAN**

Function: For editing custom code and presentation stores, this command ends the participation of a member in the current multiple member editing session, without saving the edits you made in that member for that session. Enter this command in the line number field of the member line header for the member to be canceled. You can enter this command for as many members as you are editing.

COLS**Syntax:** **COLS** | **COL**

Function: Displays a column number line to identify the column location of one or more characters below the line. Cancel this command with the RESET COLS primary command.

COMMENT**Syntax:** **COM**[*n*]

Function: (Custom code editor only.) Inserts comment lines into the member that you are currently editing, providing lines in the member into which you can enter comments. You can override the default three lines provided by specifying *n* as the number of lines.

In COBOL members, an asterisk at the beginning of the line identifies each comment line. In PL/I members, the comment lines are delimited with the PL/I comment characters /* and */ located in columns 2-3 and 70-71. In JCL, the comment lines are delimited with the JCL comment characters /**.

COPY**Syntax:** **C**[*n*] | **CC** ... **CC**

Function: Copies a line or a block of lines to another location.

To mark a single line to copy, enter **C** on the line.

To mark more than one line to copy, do one of the following:

- Enter **C***n* on the first line to copy, where *n* is the number of consecutive lines, including the first one, to be copied
- Enter **CC** on both the first and last line of the block to be copied

Note: For editing a panel you cannot COPY OVER any lines containing fields in a way that would require separating the fields

For editing custom code or presentation stores, *Cn* or *CC* are not valid on a member line.

Specify the location of the copy with AFTER, BEFORE, or OVER. AFTER and BEFORE cause the lines to be inserted. OVER causes the lines to overlay existing lines.

You can use COPY with the CREATE and REPLACE primary commands.

Note: Excessive simultaneous multiple entry of this command will cause the editor's performance to degrade.

DEFINE GROUP

Syntax: G

Function: (Panel editor only.) Creates a batch group from panel image and/or panel definition data. The editor creates a line on the screen called a batch group header that specifies the length of the batch group. Also, use the UPDATE GROUP line command to specify the name and type of the batch group.

DELETE

Syntax: D[n] | DD ... DD

Function: Deletes a line or a block of lines.

To mark a single line to delete, enter **D** on the line.

To mark more than one line to delete, do one of the following:

- Enter **Dn** on the first line to delete, where *n* is the number of consecutive lines, including the first one, to be deleted
- Enter **DD** on both the first and last line of the block to be deleted

Note: Delete SEGLOOP information by entering **D** on the start or end line of the SEGLOOP.

Deletes will not cross member boundaries.

DELETE GROUP

Syntax: **DG**

Function: (Panel editor only.) Deletes an entire batch group created with the DEFINE GROUP command. Enter **DG** in the number field of the batch group header line.

Note: To delete the *last* batch group in a panel:

1. Transfer from the panel to the Panel Definition menu
2. Enter **PU** (purge) in the FUNCTION field for the panel
3. After returning to the panel, enter the PURGE primary command

END

Syntax: **END**

Function: For custom code and presentation stores, this command ends the participation of a member in the current multiple member editing session, saving the edits that you made in the member for that session. Enter this command in the line number field of the member line header for the member to be ended. You can enter this command for as many members as you are editing.

EXCLUDE

Syntax: **X[n] | XX ... XX**

Function: Ends display of a line or a block of lines without deleting.

To mark a single line to exclude, enter **X** on the line.

To mark more than one line to exclude, do one of the following:

- Enter **Xn** on the first line to exclude, where *n* is the number of consecutive lines, including the first one, to be excluded
- Enter **XX** on both the first and last line of the block to be excluded

For editing custom code, this command may not be entered on a member line.

To cancel this command, use the FIRST or LAST line command, or the RESET primary command.

Note: When one or more lines are excluded from the Data Group and the Create/Update Data Group screen is saved on exit, the exclusion is retained. The next time the Data Group is displayed with the exclusion, issue the "RESET" command to remove the exclusion.

LAST

Syntax: `L[n]`

Function: Redisplays lines excluded with the EXCLUDE line command, beginning with the last excluded line and redisplaying *n* consecutive lines.

For editing custom code, this command may not be entered on a member line.

LEFT SHIFT

Syntax: `([n] | (([n] ... ((`

Function: Shifts the text in a line or a block of lines to the left.

To mark a single line to shift two spaces to the left, enter `(` on the line. To override the default 2, enter `(n` where *n* is the number of spaces to shift.

To mark a block of lines to shift left, enter `((n` on the first line and `((` on the last line of the block to be shifted, where *n* is the number of spaces to shift, overriding the default 2.

For editing custom code:

- This command may not be entered on a member line
- If you shift text off the side of the screen, that text will be lost

LINE CLEAR

Syntax: `LC`

Function: (Panel editor only.) Cancels the panel definition(s) associated with the field(s) in the line on which you enter the command.

MOVE

Syntax: `M[n] | MM ... MM`

Function: Moves a line or a block of lines to another location.

To mark a single line to move, enter `M` on the line.

To mark more than one line to move, do one of the following:

- Enter `Mn` on the first line to move, where *n* is the number of consecutive lines, including the first one, to be moved
- Enter `MM` on both the first and last line of the block to be moved

Specify the location of the move with AFTER, BEFORE, or OVER:

- For editing a panel, AFTER and BEFORE cause the lines to be inserted, and OVER causes the lines to overlay existing lines. However, you cannot MOVE OVER any lines containing fields in a way that would require separating the fields.
- For editing custom code, AFTER and BEFORE cause the lines to be inserted, and OVER causes the lines to overlay existing lines. However, if you use the OVER command, the editor deletes moved lines from their original locations only if the overlaid lines do not contain data in such a way as to prevent the move from succeeding. If you move an entire member, only the *lines* within the member are moved, thus leaving a custom code member that is empty. This command is not valid on a member line.
- For editing a presentation store, AFTER and BEFORE cause the lines to be inserted, and OVER causes the lines to overlay existing lines. If you move an entire member, only the *lines* within the member are moved, thus leaving a presentation store. This command is not valid on a member line.

You can use MOVE with the CREATE and REPLACE primary commands.

Note: Excessive simultaneous multiple entry of this command will cause the editor's performance to degrade.

OVER

Syntax: **O**[*n*] | **OO** ... **OO**

Function: Specifies the location of a line or block of lines at which to begin overlaying lines marked for copy or move, in conjunction with the COPY or MOVE line command.

For editing a panel, you cannot COPY OVER or MOVE OVER any line containing fields in a way that would require separating the fields.

REPEAT

Syntax: **R**[*n*] | **RR**[*n*] ... **RR**

Function: Repeats a line or a block of lines one or *n* times.

To mark a single line to repeat, enter **R** or **Rn** on the line.

To mark a block of lines to repeat, enter **RRn** on the first line and **RR** on the last line of the block to be repeated, where *n* is the number of times the block is to be repeated.

For editing custom code, if you type **R***n* or **RR***n*, press SPACEBAR or EOF before pressing Enter.

Note: The value of *n* may not exceed 50.

Excessive simultaneous multiple entry of this command will cause the editor's performance to degrade.

RESET LINE

Syntax: RESET

Function: (Panel editor only.) In line edit mode, this command cancels your most recent updates to a line containing PDs and displays the line as it was before those updates.

RESTORE

Syntax: REST

Function: For editing custom code or presentation stores, this command cancels all edits made to a member during an editing session since the last SAVE command, and restores the member to the state it was in when you issued the SAVE command (or when the editing session began). Enter this command in the line number field of the member line header for the member to be ended. You can enter this command for as many members as you are editing.

RIGHT SHIFT

Syntax:) [*n*] |)) [*n*] ...))

Function: Shifts the text in a line or a block of lines to the right.

To mark a single line to shift two spaces to the right, enter **)** on the line. To override the default 2, enter **)***n* where *n* is the number of spaces to shift.

To mark a block of lines to shift right, enter **))***n* on the first line and **))** on the last line of the block to be shifted where *n* is the number of spaces to shift, overriding the default 2.

For editing custom code:

- This command may not be entered on a member line
- If you shift text off the side of the screen, that text will be lost

SAVE**Syntax: SAVE**

Function: For editing custom code or presentation stores, this command saves the edits made to a member during the current editing session. Enter this command in the line number field of the member line header for the member to be ended. You can enter this command for as many members as you are editing.

UPDATE GROUP**Syntax: U**

Function: (Panel editor only.) Updates information for a batch group created with the DEFINE GROUP command.

On the Update Panel Fields (Batch) screen, enter **U** in the line number field of the batch group's HEADER line. This transfers you to the Update Panel Group screen, where you can enter information on the batch group.

CA-Telon Command Summary

The primary and line command tables in this section are in alphabetical order. Each table contains a column for each editor; a bullet (•) in the column means the command is valid for the editor.

Primary Commands

Primary Command	Panel Editor		Custom Code	Presentation Store	Panel Definition Editor
	Full Screen	Line Edit			
BACKWARD	•	•	•	•	•
CANCEL	•	•	•	•	•
CAPS	•	•	•	•	•
CH		•			
CHANGE			•	•	
COPY		•	•	•	
CREATE		•	•	•	
EDIT			•	•	
END	•	•	•	•	•
END HOLD	•	•	•	•	•

EQUATE		■		■		■	
FIND				■		■	
FORWARD	■		■		■		■
HELP	■		■		■		■
HOLD	■		■		■		■
ISPF	■		■		■		■
LEFT	■		■				■
LINE EDIT OUT	■		■				■
LOCATE		■		■		■	
MAXLINE		■					
MBCOPY		■		■			
MBMOVE		■		■			
MBPURGE		■		■			
MENU	■		■		■		■
MERGE		■					
MOVE	■		■		■		
NOSHOW							■
NULLS	■		■		■		■
PANEL COLUMNS		■					
PANEL LINES		■					
PANEL SIZE		■					
PD	■		■				
PDF	■		■		■		■
PDS COPY				■			
POSITION		■					■
PROFILE	■		■		■		
PURGE	■		■		■		
REPEAT CHANGE				■			
REPEAT FIND				■		■	

REPLACE		■	■	■	
RESET	■	■	■	■	■
RESTORE	■	■	■	■	
RESTORE FIND			■	■	
RESUME	■	■	■	■	■
RIGHT	■	■	■		■
SAVE	■	■	■	■	■
SHOW			■	■	■
SUBMIT			■		
SWAP	■	■	■	■	■
SWAP EDIT	■	■			■
TRANSFER		■	■	■	
TSO		■	■	■	

Line Commands

Primary Command	Panel Editor		Custom Code	Presentation Store	Panel Definition Editor
	Full Screen	Line			
AFTER		■	■	■	
AFTER MULTIPLE		■	■	■	
BEFORE		■	■	■	
BEFORE MULTIPLE		■	■	■	
BOX			■		
CANCEL			■	■	
COLS		■	■	■	

COMMENT		■		
COPY	■	■	■	
DEFINE GROUP	■			
DELETE	■	■	■	■
DELETE GROUP	■			
END		■	■	
EXCLUDE	■	■	■	
FIELD SPLIT	■			
FIRST	■	■	■	
INSERT	■	■	■	■
INSERT SPACE	■	■	■	
LAST	■	■	■	
LEFT SHIFT	■	■		
LINE CLEAR	■			
MOVE	■	■	■	
OVER	■	■	■	
REPEAT	■	■	■	
RESET LINE	■			

RESTORE

■

■

RIGHT SHIFT

■

■

SAVE

■

■

UPDATE GROUP

■

Appendix A: Field Edit Formats

CA Telon-generated programs perform field edits:

- On **output** when fields are moved from a file or work area to the screen. On output, a field edit simply reformats the field as it is moved.
- On **input** when fields are moved from the screen to a file or work area. On input, a field edit first checks the field for valid format and/or content before reformatting it for storage.

Standard field edit routines and modules

CA Telon supplies a number of standard field edit routines. This appendix documents these routines, the parameters they pass, and the processing they perform.

Customized field edit routines

You can also code your customized edit routines for use at your installation.

Specifying a field edit

In the TDF, you can specify a field edit when you create a panel definition. The value of the FLDTYPE field, appearing on TDF panel field update screens, defines the data type of the application screen field (date, dollar, state code, and so on) and determines the corresponding field edit logic called to validate or format the field.

Standard Field Edit Routines

The following identify valid FLDTYPE values associated with standard field edit routines. In these tables, the data type processed Alpha is defined as PIC X for COBOL and CHAR for PL/I. The data type Numeric is defined as PIC 9 for COBOL and PIC BINARY, or FLOAT for PL/I.

CA Telon Output Edit Routines

FLDTYPE Value	Module Name	Type	Output Edit Action
CAR	OCAR	Alpha	Replaces unprintable characters with a period.
CART	OCART	Alpha	Replaces unprintable characters with a period.

FLDTYPE Value	Module Name	Type	Output Edit Action
CJULIAN	OCJULIAN OINTCJUL	Numeric	<p>(INTDATE=U) Converts a date stored in <i>ccyyddd</i> format to the U.S. format of <i>mm/dd/ccyy</i> or <i>mmddccyy</i>, depending on the output field length.</p> <p>(INTDATE=I) Generates a call to OINTCJUL that converts a date stored in <i>ccyyddd</i> format to the international format of <i>dd/mm/ccyy</i> or <i>ddmmccyy</i>, depending on the output field length.</p> <p>Note: INTDATE is a parameter in TLNIIS, and the default is U.</p>
CDATE	OCDATE	Numeric	<p>(INTDATE=U) Converts a date stored in <i>ccyyymmdd</i> format to the U.S. format of <i>mm/dd/ccyy</i> or <i>mmddccyy</i>, depending on the output field length.</p> <p>(INTDATE=I) Generates a call to OINTCDT that converts a date stored in <i>ccyyymmdd</i> format to the international format of <i>dd/mm/ccyy</i> or <i>ddmmccyy</i>, depending on the output field length.</p> <p>Note: INTDATE is a parameter in TLNIIS, and the default is U.</p>
DATE	ODATE OINTLDT	Numeric	<p>(INTDATE=U) Converts a date stored in <i>yymmdd</i> format to the U.S. format of <i>mm/dd/yy</i> or <i>mmddyy</i>, depending on the output field length.</p> <p>(INTDATE=I) Generates a call to OINTLDT that converts a date stored in <i>yymmdd</i> format to the international format of <i>dd/mm/yy</i> or <i>ddmmyy</i>, depending on the output field length.</p> <p>Note: INTDATE is a parameter in TLNIIS, and the default is U.</p>
FLNULL	OFLNULL	SQLNULL	If the field is null, spaces are moved to TPO-FLDNAME. Otherwise, the data is processed as the OFLOAT field edit does.
FLOAT	OFLOAT OFORMAT	Numeric	Right-justifies a floating-point or signed-numeric field with an implied decimal point, using the smallest character representation possible with no loss of digits to the left of the (implied) decimal point.
HEX	OHEX	Alpha	Converts an EBCDIC character format item to hexadecimal.
HEXA	OHEXA	Alpha	Converts an EBCDIC character format item to hexadecimal.
HEXP	OHEXP	Alpha	Converts an EBCDIC character format item to hexadecimal.

FLDTYPE Value	Module Name	Type	Output Edit Action
HEXT	OHEXT	Alpha	Converts an EBCDIC character format item to hexadecimal.
JULIAN	OJULIAN OINTJUL	Numeric	<p>(INTDATE=U) Converts a date stored in <i>yyddd</i> format to the U.S. format of <i>mm/dd/yy</i> or <i>mmddyy</i>, depending on the output field length.</p> <p>(INTDATE=I) Generates a call to OINTLJUL that converts a date stored in <i>yyddd</i> format to the international formats <i>dd/mm/yy</i> or <i>ddmmyy</i>, depending on the output field length.</p> <p>Note: INTDATE is a parameter in TLNIIS, and the default is U.</p>
NULL	ONULL	SQLNULL	If the field is null, spaces are moved to TPO-FLDNAME. Otherwise, the output field name is moved to TPO-FLDNAME.
NUMNULL	OFLNULL	ONUMNULL	If the field is null, spaces are moved to TPO-FLDNAME. Otherwise, TPO-FLDNAME is formatted using the output fieldname integers, and they are right-justified and blank-filled.
SSA	OSSA	Alpha	(DL/I processing only.) Parses an IMS SSA to determine the input SSA length. Returns both the SSA length and the SSA itself. Specified in the PIC field of the Update Output/Input/Outin Field screen.
VCHAR	OVCHAR	Variable character	Maps data from DBNAME <i>hvname</i> to TPO- <i>fldname</i> , where <i>hvname</i> is defined for the SQL VARCHAR column.
VNULL	OVNULL	SQLNULL	If the field is null, spaces are moved to TPO-FLDNAME. Otherwise, the data is mapped from DBNAME <i>hvname</i> to TPO-FLDNAME, where <i>hvname</i> is defined for SQL VARCHAR column.

CA Telon Input Edit Routines

FLDTYPE Name	Module Value	Type	Input Edit Action
CDATE	ICDATE IINTCDT	Numeric	<p>(Set <i>INTDATE</i>=U) Edits a date field for a valid U.S. format of <i>mm/dd/ccyy</i>, <i>mm-dd-ccyy</i>, or <i>mmddccyy</i>, depending on the input field length; then converts it to <i>ccyyymmdd</i> format.</p> <p>(<i>INTDATE</i>=I) Generates a call to IINTCDT that edits a date field for a valid International format of <i>dd/mm/ccyy</i>, <i>dd-mm-ccyy</i>, or <i>ddmmccyy</i>, depending on the input field length; then converts it to <i>ccyyymmdd</i> format.</p> <p>Note: <i>INTDATE</i> is a parameter in TLNIIS and the default is U.</p>
CJULIAN	ICJULIAN IINTCJUL	Numeric	<p>(Set <i>INTDATE</i>=U) Edits a date field for a valid U.S. format of <i>mm/dd/ccyy</i>, <i>mm-dd-ccyy</i>, or <i>mmddccyy</i>, depending on the input field length; then converts it to <i>ccyyddd</i> format.</p> <p>(<i>INTDATE</i>=I) Generates a call to IINTCJUL that edits a date field for a valid International format of <i>dd/mm/ccyy</i>, <i>dd-mm-ccyy</i>, or <i>ddmmccyy</i>, depending on the input field length; then converts it to <i>ccyyddd</i> format.</p> <p>Note: <i>INTDATE</i> is a parameter in TLNIIS and the default is U.</p>

FLDTYPE Name	Module Value	Type	Input Edit Action
DATE	IDATE IINTLDT	Numeric	<p>(Set <i>INTDATE=U</i>) Edits a date field for a valid U.S. format of mm/dd/yy, mm-dd-yy or mmddyy, depending on the input field length; then converts it to yymmdd format.</p> <p>(<i>INTDATE=I</i>) Generates a call to IINTLDT that edits a date field for a valid International format of dd/mm/yy, dd-mm-yy, or ddmmyy, depending on the input field length; then converts it to yymmdd format.</p> <p>Note: INTDATE is a parameter in TLNIIS and the default is U.</p>
DOLLAR	IDOLLAR IBDOL ICURRENCY	Numeric	<p>(<i>INTDATE=U</i>) Edits a field for a valid U.S. dollar and cents format (that is, two decimal places preceded by a period). This field does not accept negatives.</p> <p>(<i>INTDATE=I</i>) Generates a call to ICURRENCY that edits a field for a valid International currency format (that is, two decimal places preceded by a comma).</p> <p>INTDATE is a parameter in TLNIIS and the default is U.</p>
FLNULL	IFLNULL	SQLNULL	If completely blank on input, FLNULL sets the null indicator. Otherwise, FLNULL edits a field for numeric format, allowing for a decimal point and a leading sign.
FLOAT	IFLOAT	Numeric	Edits a field for valid numeric format, allowing for a decimal point and leading sign.
FULLCAR	IFULLCAR	Alpha	Edits an input field to ensure there are no blanks in the field.
FULLNUM	IFULLNUM	Numeric	Edits a numeric field to ensure that all characters are numeric.

FLDTYPE Name	Module Value	Type	Input Edit Action
JULIAN	IJULIAN	Numeric	<p>(INTDATE=U) Edits a date field for a valid U.S. format of <i>mm/dd/yy</i>, <i>mm-dd-yy</i>, or <i>mmddy</i>, depending on the input field length; then converts it to <i>yyddd</i> format.</p> <p>(INTDATE=I) Generates a call to IINTLJUL that edits a date field for a valid International format of <i>dd/mm/yy</i>, <i>dd-mm-yy</i>, or <i>ddmmyy</i>, depending on the input field length; then converts it to <i>yyddd</i> format.</p> <p>Note: INTDATE is a parameter in TLNIIS and the default is U.</p>
LALPHA	ILALPHA	Alpha	Left-justifies a character field.
LNULL	ILNULL	SQLNULL	If blank on input, LNULL sets the null indicator. Otherwise, LNULL left-justifies a character field.
LVCHAR	ILVCHAR	Variable character	Left-justifies a character field (<i>hvname</i> is defined for SQL VARCHAR column).
LVNULL	ILVNULL	SQLNULL	If blank on input, LVNULL sets the null indicator. Otherwise, LVNULL left-justifies a variable-length character field into DBNAME <i>hvname</i> (<i>hvname</i> is defined for the SQL VARCHAR column).
NBALPHA	INBALPHA	Alpha	Edits a field to ensure there are no embedded blanks. NBALPHA allows leading and/or trailing blanks; FULLCAR does not.
NBNULL	INBNULL	SQLNULL	If blank on input, NBNULL sets the null indicator. Otherwise, NBNULL edits a field to ensure there are no embedded blanks.
NBVCHAR	INBVCHAR	Variable character	Edits a field to ensure there are no embedded blanks. (<i>hvname</i> is defined for SQL VARCHAR column.)
NBVNULL	INBVNULL	SQLNULL	If blank on input, NBVNULL sets the null indicator. Otherwise, NBVNULL edits a variable-length character field to ensure there are no embedded blanks.
NULL	INULL	SQLNULL	If blank on input, NULL sets the null indicator. Otherwise, NULL maps data from TPI-FLDNAME to DBNAME.
NUMERIC	INUMERIC	Numeric	Edits a field for valid numeric format, allowing for leading or trailing blanks (or both).
NUMNULL	INUMNULL	SQLNULL	If blank on input, NUMNULL sets the null indicator. Otherwise, NUMNULL edits a field for valid numeric

FLDTYPE Name	Module Value	Type	Input Edit Action
			format, allowing for leading and/or trailing spaces.
TATE	ISTATE	Alpha	Edits a field for valid two-character postal state code.
VCHAR	IVCHAR	Variable character	Maps data from TPO- <i>fldname</i> to DBNAME <i>hvname</i> (<i>hvname</i> is defined for SQL VARCHAR column).
VNULL	IVNULL	SQLNULL	If blank on input, VNULL sets the null indicator. Otherwise, VNULL maps data from TPI- <i>fldname</i> to DBNAME <i>hvname</i> (<i>hvname</i> is defined for the SQL VARCHAR column.)

Standard Field Edit Modules

The following table lists standard field edit modules provided by CA Telon and summarizes the edit function associated with each module.

Detailed documentation of each call module is provided after the table that includes:

- A description
- Input parameters
- Output parameters
- Edit processing

See 1 for more information about edit processing.

Module Name	Edit Function
OCAR	Replace unprintable characters with periods
OCART	Replace unprintable characters with periods
OCDATE	Format century date
OCJULIAN	Convert century Julian date to <i>mm/dd/ccyy</i>
ODATE	Format date
OFLNULL	Format a floating-point field that may be null
OFLOAT	Format a floating-point field
OFORMAT	Reformat for output
OHEX	Convert character to hex

Module Name	Edit Function
OHEXA	Convert characters to hex representation
OHEXP	Convert characters to hex representation
OHEXT	Convert characters to hex representation
OINTCDT	Format century date (international)
OINTCJUL	Format century Julian date (international)
OINTLDT	Format date (international)
OINTLJUL	Format Julian date (international)
OJULIAN	Convert Julian date to <i>mm/dd/yy</i>
ONULL	Format field based on null indicator
ONUMNULL	Format integer field based on null indicator
OSSA	Determine length of SSA
OVCHAR	Alphanumeric move from variable-length character field
OVNULL	Alphanumeric move from variable-length character field
IBDOL	Validate dollar entry
IBCURRE	Validate international currency entry
ICURRNCY	Validate international currency entry
IBNUM	Validate numeric entry
ICDATE	Check for valid century date
ICJULIAN	Convert century date to century Julian
IDATE	Check for valid date
IDOLLAR	Check for dollar/currency entry
IFLNULL	Validate floating-point entry and set null indicator
IFLOAT	Check for floating-point number
IFULLCAR	Check for blanks
IFORMAT	Validate and reformat on input
IFULLNUM	Check for all numeric characters
IHEX	Convert hex representation to hex characters
IHEXA	Convert hex representation to hex characters
IINTCDT	Validate century date (international)
IINTCJUL	Convert century date to century Julian (international)

Module Name	Edit Function
IINTLDT	Validate date (international)
IINTLJUL	Validate date (international)
IJULIAN	Convert date to Julian
ILALPHA	Left-justify a character field
ILNULL	Set null indicator and left-justify character field
ILVCHAR	Left-justify a variable-length character field
ILVNULL	Left-justify a variable-length length field and set null indicator
INBALPHA	Check for imbedded blanks
INBNULL	Check for imbedded blanks and set null indicator
INBVCHAR	Check for imbedded blanks
INBVNULL	Check blanks for variable-length field and set null indicator
INULL	Set null indicator
INUMERIC	Check for valid integer
INUMNULL	Check for valid integer and set null indicator
ISTATE	Check for valid state code
IVCHAR	Alphanumeric move to variable-length character field
IVNULL	Set null indicator for variable-length field

OCAR

Function

Replaces unprintable characters with periods.

Parameters

Type	Name	Description
Input	TPO- <i>fieldname</i> -LTH	Length of the edited field. This must be between 1 and 256.
Input	<i>output-fieldname</i>	Field to be edited by this routine.
Output	TPO- <i>fieldname</i>	Edited output of the routine that is returned to the TPO buffer.

Edit processing

1. Search the output field for unprintable characters.
2. Translate unprintable characters to periods (X'4B').

OCART**Function**

Replaces unprintable characters with periods. The difference between this field edit and IHEX is that this edit is passed the address of the field to be converted, rather than the value of the field itself.

Parameters

Type	Name	Description
Input	TPO- <i>fieldname</i> -LTH	Length of the edited field. This must be between 1 and 256.
Input	<i>output-fieldname</i>	Field to be edited by this routine.
Output	TPO- <i>fieldname</i>	Edited output of the routine that is returned to the TPO buffer.

Edit processing

1. Search *output-fieldname* for unprintable characters.
2. Translate unprintable characters to periods (X'4B').

OCDATE**Function**

Format a date stored as *ccyyymmdd* for output as either *mm/dd/cc/yy* (if TPO-*datefield*-LTH is 10) or *mmddccyy* (if the TPO-*datefield*-LTH is 8).

Note: If the INTDATE parameter in the TLNIIS macro is set to I, a call to OINTCDT is generated in place of a call to ODATE. For more information, refer to OINTCDT.

Parameters

Type	Name	Description
Input	WORKFLD-NUMERIC	Date to be edited. In a COBOL PIC

Type	Name	Description
		S9(11)V9(7) field, the date is laid out as 000ccyyymmdd00000000. In a PL/I PIC 10(9)V(4)9T field, it is laid out as 00ccyyymmdd000000.
Input	TPO- <i>datefield</i> -LTH	Length of the output field.
Output	TPO- <i>datefield</i>	Edited date returned to the TPO buffer.

Edit processing

1. Move ccyy in WORKFILED-NUMERIC to make the result *mmddccyy*.
2. For 10-character output, insert the slashes.
3. Move the edited form to the output field.

OCJULIAN

Function

Formats a date stored in *ccyyddd* for output as *mm/dd/ccyy* (if TPO-*datefield*-LTH is 10) or *mmddccyy* (if TPO-*datefield*-LTH is 8).

Note: If the INTDATE parameter in the TLNIIS macro is set to 'I,' a call to OINTCJUL is generated in place of a call to OCJULIAN. See OINTCJUL for more information.

Parameters

Type	Name	Description
Input	WORKFLD-NUMERIC	Date to be edited. In a COBOL PIC S9(11)V9(7) field, the date is laid out as 0000ccyyddd00000000. In a PL/I PIC (10)9V(4)9T field, it is laid out as 000ccyyddd000000.
Input	TPO- <i>fieldname</i> -LTH	Length of the output field.
Output	TPO- <i>fieldname</i>	Converted output of the routine that is returned to the TPO buffer.

Edit processing

1. Convert *ddd* from the *ccyyddd* of WORKFLD-NUMERIC to *mmdd*.

Note: If *ddd* is greater than 365 (366 for leap year), *yy* is adjusted one year for each additional 365 (366) days.

2. If TPO-*fieldname*-LTH is 10, insert slashes (/) into the *mmddccyy* to create the *mm/dd/ccyy* format.
3. Move the reformatted value to TPO-*fieldname*.

ODATE

Function

Formats a date stored as *yymmdd* for output as either *mm/dd/yy* (if TPO-*datefield*-LTH is 8) or *mmddyy* (if the TPO-*datefield*-LTH is 6).

Note: If the INTDATE parameter in the TLNIIS macro is set to 'I', a call to OINTLDT is generated in place of a call to ODATE. For more information, see 33.

Parameters

Type	Name	Description
Input	WORKFLD-NUMERIC	Date to be edited. In a COBOL PIC S9(11)V9(7) field, the date is laid out as 00000yymmdd0000000. In a PL/I (10)9V(4)9T field, it is laid out as 0000yymmdd00000.
Input	TPO- <i>datefield</i> -LTH	Length of the output field.
Output	TPO- <i>datefield</i>	Edited date returned to the TPO buffer.

Edit processing

1. Move *yy* in WORKFLD-NUMERIC to make the result *mmddyy*.
2. For eight-character output, insert the slashes.
3. Move the edited form to the output field.

OFLNULL

Function

Right-justifies a floating-point or signed-numeric field.

If the null indicator is set, the output field is set to spaces. The edit uses the minimum number of characters required to include all significant digits to the left of the implied decimal point. Then, given the output field length, it formats as many significant digits to the right of the decimal point as possible. Extra digits to the right of the decimal are truncated, not rounded.

The routine signifies an overflow condition (the output field is too small to contain the whole number) by filling the output field with all asterisks (*).

Parameters

Type	Name	Description
Input	WORKFLD-NUMERIC	Numeric value from the DBNAME field to be edited. In COBOL, PIC S9(11)V9(7); in PL/I, PIC (10)9V(7)9T.
Input	TPO- <i>fieldname</i> -LTH	Length of the output field. It must be 1 to 18 bytes for COBOL, 1 to 15 for PL/I.
Input	WK- <i>fieldname</i> -NV	Null indicator variable, set to -1 by SQL when the DBNAME is null.
Output	TPO- <i>fieldname</i>	Edited field returned to the TPO output buffer.

Edit processing

1. Determine if the field is null by checking the null indicator.
2. If the field is null, move spaces to the TPO buffer field. Otherwise, if WORKFLD-NUMERIC is negative, store a minus sign in the first position of TEMP-OUT-FIELD, a temporary, internal, alphanumeric work area.
3. Bypass any leading zeros encountered before the implied decimal.
4. Find any significant digits to the right of the decimal point and store them in TEMP-OUT-FIELD after any minus sign.
5. If the significant positions after the implied decimal are all zeros, exit the edit processing. Otherwise, store a decimal point in TEMP-OUT-FIELD following the digits already there.
6. Check to see if there are more significant characters in the TEMP-OUT-FIELD than specified by TPO-*fieldname*-LTH. If so, fill TPO-*fieldname* with all asterisks (*). Then exit the edit processing.

- Find all remaining digits after the decimal and up to the last group of all zeros. Store these digits in TEMP-OUT-FIELD after the decimal point.
- Right-justify the TEMP-OUT-FIELD value in TPO-*fieldname* and end the routine.

OFLOAT

Function

Right-justify a floating-point or signed-numeric field.

The edit uses the minimum number of characters required to include all significant digits to the left of the implied decimal point. Then, given the output field length, it formats as many significant digits right of the decimal point as possible. Extra digits right of the decimal are truncated, not rounded.

The routine signifies an overflow condition by filling the output field with all plus (+) or minus (-) signs, depending on the sign of the input value.

Parameters

Type	Name	Description
Input	WORKFLD-NUMERIC	Value to be edited. Before editing, the value is moved to this field from the DBNAME field to be output. In COBOL, PIC S9(11)V9(7); in PL/I, PIC (10)9V(4)9T.
Input	TPO- <i>fieldname</i> -LTH	Length of the edited field. It must be from one to 18 bytes.
Output	TPO- <i>fieldname</i>	Edited field returned to the TPO output buffer.

Edit processing

- If WORKFLD-NUMERIC is negative, store a minus sign in the first position of TEMP-OUT-FIELD, a temporary, internal, alphanumeric work area.
- Bypass any leading zeros encountered before the implied decimal.
- Find any significant digits right of the decimal point and store them in TEMP-OUT-FIELD after any minus sign.
- If the significant positions after the implied decimal are all zeros, exit the edit processing. Otherwise, store a decimal point in TEMP-OUT-FIELD following the digits already there.

5. Check to see if there are more significant characters in TEMP-OUT-FIELD than specified by TPO-*fieldname*-LTH. If so, fill TPO-*fieldname* with all plus (+) or all minus (-) signs, depending on the sign of the input field. Then exit the edit processing.
6. Find all remaining digits after the decimal up to the last group of all zeros and store these in TEMP-OUT-FIELD after the decimal point.
7. Right-justify TEMP-OUT-FIELD value in TPO-*fieldname* and end the routine.

OFORMAT

Function

Reformats output.

Specification for this special edit is made by entering the FORMAT command in the SPEC field on the Update Output/Input/Outin Field screen or the Update Batch Output Fields screen. See Update Output/Input/Outin Field for more information.

OHEX

Converts a character string to the hexadecimal equivalent for each character.

Parameters

Type	Name	Description
Input	<i>output-fieldname</i>	Field holding the characters to be converted to hex.
Input	TPO- <i>fieldname</i> -LTH	Length of the edited field. It can be from 2 to 256 bytes long.
Output	TPO- <i>fieldname</i>	Edited field returned to the TPO buffer.

Edit processing

1. Determine the output field length.
2. Use the assembly language TRANSLATE instruction and convert the input in eight-byte pieces.
3. If fewer than eight characters remain, convert the last bytes individually.

OHEXA

Function

Converts a one- to four-character string to the hexadecimal equivalent for each character (for example, 'AB12' converts to 'C1C2F1F2'). This field edit is designed specifically for converting addresses.

Parameters

Type	Name	Description
Input	TPO- <i>fieldname</i> -LTH	Length of the edited field. This must be between 1 and 256.
Input	<i>output-fieldname</i>	Field to be converted by this routine.
Output	TPO- <i>fieldname</i>	Converted output of the routine that is returned to the TPO buffer.

Edit processing

1. Determine the output field length.
2. Use the assembler language TRANSLATE instruction to convert the output.
3. If the output length is not equal to 8, the first byte of input is ignored, and bytes 2 through 4 are converted (with the assumption that the input is a 24-bit address). Otherwise, all four bytes are converted.

OHEXP

Function

Converts a one- to four-character string to the hexadecimal equivalent for each character (for example, 'AB12' converts to 'C1C2F1F2'). The difference between this field edit and OHEX is that this edit is passed the address of the field to be converted, rather than the value of the field itself.

Parameters

Type	Name	Description
Input	TPO- <i>fieldname</i> -LTH	Length of the edited field. This must be between 1 and 256.
Input	<i>output-fieldname</i>	Field to be converted by this routine.
Output	TPO- <i>fieldname</i>	Converted output of the routine that is

Type	Name	Description
		returned to the TPO buffer.

Edit processing

1. Determine the output field length.
2. Use the assembler language TRANSLATE instruction to convert the output.
3. Move the output to TPO-*fieldname*.

OHEX**Function**

Converts a one- to four-character string to the hexadecimal equivalent for each character (for example, 'AB12' converts to 'C1C2F1F2'). The difference between this field edit and OHEX is that this edit is passed the address of the field to be converted, rather than the value of the field itself.

Parameters

Type	Name	Description
Input	TPO- <i>fieldname</i> -LTH	Length of the edited field. This must be between 1 and 256.
Input	<i>output-fieldname</i>	Field to be converted by this routine.
Output	TPO- <i>fieldname</i>	Converted output of the routine that is returned to the TPO buffer.

Edit processing

1. Determine the output field length.
2. Use the assembler language TRANSLATE instruction to convert the output.
3. Move the output to TPO-*fieldname*.

OINTCDT

Function

Reformats a date stored in *ccyyymmdd* to *dd/mm/ccyy* (if *TPO-datefield-LTH* is 10) or to *ddmmccyy* (if the *TPO-datefield-LTH* is 8).

A call to this field is generated in either of the following cases:

- It is requested directly (that is, 'INTCDT' is identified as the output field edit)
- The INTDATE parameter in the TLNIIS macro is set to 'I' and DATE is identified as the output field edit

Parameters

Type	Name	Description
Input	WORKFLD-NUMERIC	Date to be edited. In a COBOL PIC S9(11)V9(7) field, the date is laid out as 000ccyyymmdd0000000. In a PL/I PIC (10)9V(4)9T field, it is laid out as 00ccyyymmdd000000.
Input	TPO- <i>datefield-LTH</i>	Length of the output field.
Output	TPO- <i>datefield</i>	Converted output of the routine that is returned to the TPO buffer.

Edit processing

1. Determine the output field length.
2. Reformat the WORKFLD-NUMERIC date value from *ccyyymmdd* to *ddmmccyy*. If *TPO-datefield-LTH* is 10, insert slashes (/) to produce *dd/mm/ccyy*.
3. Move the reformatted value to *TPO-datefield*.

OINTCJUL

Function

Reformats a date stored in *ccyyddd* format to *dd/mm/ccyy* (if *TPO-datefield-LTH* is 10) or *ddmmccyy* (if *TPO-datefield-LTH* is 8).

A call to this field edit is generated in either of these cases:

- It is requested directly (that is, 'INTCDT' is identified as the output field edit)
- The INTDATE parameter in the TLNIIS macro is set to 'I' and JULIAN is identified as the output field edit

Parameters

Type	Name	Description
Input	WORKFLD-NUMERIC	Date to be edited. In a COBOL PIC S9(11)V9(7) field, the date is laid out as 0000ccyyddd0000000. In a PL/I PIC (10)9V(4)9T field, it is laid out as 000ccyyddd000000.
Input	TPO-datefield-LTH	Length of the output field.
Output	TPO-datefield	Converted output of the routine that is returned to the TPO buffer.

Edit processing

1. Determine the output field length.
2. Reformat the WORKFLD-NUMERIC date value from *ccyyddd* to *ddmmccyy*. If *TPO-datefield-LTH* is 10, insert slashes (/) to produce *dd/mm/ccyy*.
3. Move the reformatted value to *TPO-datefield*.

Note: If *ddd* is greater than 365 (366 for leap year), *yy* is adjusted one year for each additional 365 (366) days.

OINTLDT

Function

Reformats a date stored in *yymmdd* format to *dd/mm/yy* (if TPO-*datefield*-LTH is 8) or *ddmmyy* (if TPO-*datefield*-LTH is 6).

A call to this field edit is generated in either of these cases:

- It is requested directly (that is, 'INTLDT' is identified as the output field edit)
- The INTDATE parameter in TLNIIS is set to 'I' and DATE is identified as the output field edit

Parameters

Type	Name	Description
Input	WORKFLD-NUMERIC	Date to be edited. In a COBOL PIC S9(11)V9(7) field, the date is laid out as 00000yymmdd0000000. In a PL/I PIC(10)9V(4)9T field, the date is laid out as 0000yymmdd00000.
Input	TPO- <i>datefield</i> -LTH	Length of the output field.
Output	TPO- <i>datefield</i>	Converted output of the routine that is returned to the TPO buffer.

Edit processing

1. Determine the output field length.
2. Reformat the WORKFLD-NUMERIC date value from *yymmdd* to *ddmmyy*. If TPO-*datefield*-LTH is 8, insert slashes to produce *dd/mm/yy*.
3. Move the reformatted value to TPO-*datefield*.

OINTLJUL

Function

Reformat a date stored in *yyddd* format to *dd/mm/yy* (if the TPO-*datefield*-LTH is 8) or *ddmmyy* (if the TPO-*datefield*-LTH is 6).

A call to this field edit is generated in either of these cases:

- It is requested directly (that is, 'INTLJUL' is identified as the output field edit)
- The INTDATE parameter in TLNIIS is set to 'I' and JULIAN is identified as the output field edit

Parameters

Type	Name	Description
Input	WORKFLD-NUMERIC	Date to be edited. In a COBOL PIC S9(11)V9(7) field, the date is laid out as 000000yyddd0000000. In a PL/I PIC (10)9V(4)9T field, the date is laid out as 00000yyddd000000.
Input	TPO-datefield-LTH	Length of the output field.
Output	TPO-datefield	Converted output of the routine that is returned to the TPO buffer.

Edit processing

1. Determine the output field length.
2. Reformat the WORKFLD-NUMERIC date value from *yyddd* to *ddmmyy*. If TPO-datefield-LTH is 8, insert slashes to produce *dd/mm/yy*.
3. Move the reformatted value to TPO-datefield.

Note: If *ddd* is greater than 365 (366 for leap year), *yy* is adjusted one year for each additional 365 (366) days.

OJULIAN**Function**

Formats a date stored as *yyddd* for output as *mm/dd/yy* (if TPO-datefield-LTH is 8) or *mmddyy* (if TPO-datefield-LTH is 6).

Note: If the INTDATE parameter in the macro TLNIIS is set to 'I', a call to OINTLJUL is generated in place of a call to OJULIAN. For more information, see OINTLJUL.

Parameters

Type	Name	Description
Input	WORKFLD-NUMERIC	Date to be edited. It is the numeric value moved from the DBDNAME field. In a COBOL PIC S9(11)V9(7) field the date is laid out as 000000yyddd0000000. In a PL/I as 00000yyddd000000.
Input	TPO-fieldname-LTH	Length of the output field.

Type	Name	Description
Output	TPO- <i>fieldname</i>	Edited field returned to the TPO output buffer.

Edit processing

1. Convert *ddd* from the *yyddd* of WORKFLD-NUMERIC to *mmdd*.

Note: If *ddd* is greater than 365 (366 for leap year), *yy* is adjusted up one year for each additional 365 (or 366) days.

2. Insert slashes into the *mmddyy* to create the *mm/dd/yy* format.

ONULL

Function

Formats an output character field based on the null indicator.

Parameters

Type	Name	Description
Input	OUTPUT- <i>fieldname</i>	Field containing the characters to be output. The length of OUTPUT- <i>fieldname</i> (DBNAME) must be at least as long as TPO- <i>fieldname</i> .
Input	TPO- <i>fieldname</i> -LTH	Length of the output field.
Input	WK- <i>fieldname</i> -NN	Null indicator variable, set to -1 by SQL when OUTPUT- <i>fieldname</i> is null.
Output	TPO- <i>fieldname</i>	Edited field returned to the TPO output buffer.

Edit processing

1. Determine if the field is null by checking the null indicator.
2. If the field is null, move spaces to TPO-*fieldname*. Otherwise, OUTPUT-*fieldname* is moved to TPO-*fieldname*.

ONUMNULL

Function

Right-justify an output integer to conform with a length specified by a PIC parameter on a Field statement.

If the null indicator is set, the output field is set to spaces. Otherwise, this edit right-justifies a numeric field.

The routine signifies an overflow condition by filling the output field with asterisks (*).

Parameters

Type	Name	Description
Input	WORKFLD-NUMERIC	Numeric value from the DBNAME field to be edited. In COBOL, PIC S9(11)V9(7); in PL/I, PIC(10)9V(4)9T.
Input	TPO- <i>fieldname</i> -LTH	Length of the output field.
Input	WK- <i>fieldname</i> -NN	Null indicator variable, set to -1 by SQL when the DBNAME is null.
Output	TPO- <i>fieldname</i>	Edited field returned to the TPO output buffer.

Edit processing

1. Determine if the field is null by checking the null indicator.
2. If the field is null, move spaces to TPO-FIELDNAME. Otherwise, the numeric value from the DBNAME field (WORKFLD-NUMERIC) is right-justified, blank-filled into TPO-*fieldname*.

OSSA

Function

Determines the length of an input SSA.

Parameters

Type	Name	Description
Input	<i>segmentname</i> -SSA	IMS SSA.

Type	Name	Description
Output	TPO- <i>ssaname</i>	The SSA moved to the TPO output buffer.
Output	TPO- <i>ssaname</i> -LTH	Length of the SSA, to a maximum of 256.

Edit processing

1. Search through the SSA to check for qualification.
2. If a space is found during the search, process the SSA as qualified, then branch to compute the length below.

If, instead, a left paren is found, process the SSA as unqualified. Search until an equal number of right and left parentheses is found, or until the maximum TPO-*ssaname*-LTH (256) is reached.

3. Compute the length of the SSA and move the length to TPO-*ssaname*-LTH.
4. Move the SSA to TPO-*ssaname*.

OVCHAR

Function

Returns a variable-length character field and length.

Parameters

Type	Name	Description
Input	<i>output-fieldname</i>	Variable-length character field containing the characters to be output.
Input	TPO- <i>fieldname</i> -LTH	Length of the edited field.
Output	TPO- <i>fieldname</i>	Edited field returned to the TPO buffer.

Edit processing

1. Determine the output field length (the lesser of the variable-length character field length and TPO-*fieldname*-LTH).
2. Move the character portion of the variable-length character field to the output buffer (TPO-*fieldname*).

OVNULL

Function

Returns a variable-length character field and length by performing an alphanumeric move from a variable-length character field based on the null indicator.

If the null indicator is set, the output field is set to spaces. Otherwise, the character portion of the variable-length character field is moved to the output buffer.

Parameters

Type	Name	Description
Input	OUTPUT- <i>fieldname</i>	Variable-length character field containing the characters to be output.
Input	TPO- <i>fieldname</i> -LTH	Length of the output field.
Input	WK- <i>fieldname</i> -NN	Null indicator variable, set to -1 by SQL when DBNAME is null.
Output	TPO- <i>fieldname</i>	Edited field returned to the TPO output buffer.

Edit processing

1. Determine if the field is null by checking the null indicator.
2. If the field is null, move spaces to TPO-*fieldname*. Otherwise, determine the output length (the lesser of the variable-length character field length and TPO-*fieldname*-LTH).
3. Move the character portion of the variable-length character field to the output buffer.

IBDOL

Function

Checks input for valid dollar format, with a period (.) separating dollars and cents.

The difference between this field edit and IDOLLAR is that it allows larger numbers to be validated because the WORKFLD-NUMERIC-1 field in which the validated entry is returned is defined as PIC S9(16)V9(2) for COBOL, PIC (13)9V9T for PL/I.

Note: If the INTDATE parameter in the macro TLNIIS is set to 'I', a call to IBCURR is generated in place of a call to IBDOL. For more information, see IBCURR.

Parameters

Type	Name	Description
Input	TPI-fieldname-LTH	Length of the output field.
Input	TPI- <i>fieldname</i>	Value to be edited for valid dollar amount, from the TP input buffer.
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none">■ Spaces— No error was detected■ G121— An error was detected during the edit
Output	WORKFLD-NUMERIC-1	Validated value, in format PIC S9(16)V9(2) for COBOL, PIC (13)9V9T for PL/I.

Edit processing

1. Strip incoming TPI field of leading and trailing blanks.
2. Test the antipenultimate position for a decimal point.
3. Strip the decimal point and move the justified number to WORKFLD-NUMERIC-1.
4. Zero-fill the value in WORKFLD-NUMERIC-1.
5. Test WORKFLD-NUMERIC-1 for valid numeric value.
6. Assign FIELD-EDIT-ERROR, based on outcome of decimal point placement test and numeric validation.

IBCURRE

Function

Checks for valid international currency entry (that is, with a comma separating the last two digits of the numeric entry). The difference between this field edit and ICURRNCY is that it allows larger numbers to be validated because the WORKFLD-NUMERIC-1 field in which the validated entry is returned is defined as PIC S9(16)V9(2) for COBOL, PIC (13)9V9T for PL/I.

A call to this field edit is generated in either of these cases:

- It is requested directly (that is, 'BCUR' is identified as the OUTPUT field edit).
- The INTDATE parameter in TLNIIS is set to 'I' and 'BDOL' is identified as the output field edit.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the output field.
Input	TPI- <i>fieldname</i>	Value to be edited for valid currency amount, from the TP input buffer.
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none"> ■ Spaces— No error was detected ■ G121— An error was detected during the edit
Output	WORKFLD-NUMERIC-1	Validated value, in format PIC S9(16)V9(2) for COBOL, PIC (13)9V9T for PL/I.

Edit processing

1. Strip incoming TPI field of leading and trailing blanks.
2. Test the antipenultimate position for a comma.
3. Strip the comma and move the justified number to WORKFLD-NUMERIC.
4. Zero-fill the value in WORKFLD-NUMERIC-1.
5. Test WORKFLD-NUMERIC-1 for valid numeric value.
6. Assign FIELD-EDIT-ERROR, based on outcome of decimal point placement test and numeric validation.

IBNUM

Function

Checks a field for valid integer characteristics. The difference between this field edit and ICURRENCY is that it allows larger numbers to be validated because the WORKFLD-NUMERIC-1 field in which the validated entry is returned is defined as PIC 9(16)V9(2)T for COBOL, PIC (13)9V9T for PL/I.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the output field.
Input	TPI- <i>fieldname</i>	Value to be edited for valid integer amount, from the TP input buffer.
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none">■ Spaces— No error was detected■ G121— An error was detected during the edit
Output	WORKFLD-NUMERIC-1	Validated value, in format PIC 9(16)V9(2) for COBOL, PIC (13)9V9T for PL/I.

Edit processing

1. Strip incoming TPI field of leading and trailing blanks.
2. Justify the input entry and move it to WORKFLD-NUMERIC-1.
3. Zero-fill the value in WORKFLD-NUMERIC-1.
4. Test WORKFLD-NUMERIC-1 for valid numeric value.
5. Assign FIELD-EDIT-ERROR, based on outcome of decimal point placement test and numeric validation.

ICDATE

Function

Checks input for valid date format, *mm/dd/ccyy* or *mm-dd-ccyy* (if TPO-*datefield*-LTH is 10) or *mmddccyy* (if TPO-*datefield*-Lth is 6), and stores it as *ccyyymmdd*.

Note: If the INTDATE parameter in the TLNIIS macro is set to 'I,' a call to IINTCDT is generated in place of a call to IDATE. See 1.

Parameters

Type	Name	Description
Input	TPI- <i>datefield</i> -LTH	Length of the input field. It must be a halfword binary with a value of 8 or 10.
Input	TPI- <i>datefield</i>	Contents of the TP input buffer (the date to be edited).
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none"> ■ Spaces— No error was detected ■ ERR1— Length error ■ G123— An error was detected during processing of an eight-byte date ■ G130— An error occurred during processing of a ten-byte date
Output	WORKFLD-NUMERIC	Result of the edit, in a COBOL PIC S9(11)V9(7)9 field, the date is laid out as 000ccyyymmdd0000000 for COBOL or 00ccyyymmdd000000. The program moves this to the appropriate storage area after return from the edit routine.

Edit processing

1. If the field length is ten bytes, check for valid delimiters. If invalid delimiters are found, exit with error code G130.
2. Move the date to WORKFLD-NUMERIC and test for valid month, day, and year. If invalid, return appropriate error code. Otherwise, return edited date in WORKFLD-NUMERIC.

Note: The characters *cc* and *yy* are tested for only valid numeric characters. The characters *dd* and *mm* are edited for valid characters.

ICURRNCY

Function

Checks for valid international currency entry (that is, with a comma separating the last two digits of the numeric entry).

A call to this field edit is generated in either of these cases:

- It is requested directly (that is, 'CURRNCY' is identified as the output field edit)
- The INTDATE parameter in TLNIIS is set to 'I' and 'DOLLAR' is identified as the output field edit

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the output field.
Input	TPI- <i>fieldname</i>	Value to be edited for valid currency amount, from the TP input buffer.
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none">■ Spaces— No error was detected■ G121— An error was detected during the edit
Output	WORKFLD-NUMERIC	Validated value, in format PIC S9(11)V9(7) for COBOL, PIC (10)9V(4)9T for PL/I.

Edit processing

1. Strip incoming TPI field of leading and trailing blanks.
2. Test the antipenultimate position for a comma.
3. Strip the comma and move the justified number to WORKFLD-NUMERIC.
4. Zero-fill the value in WORKFLD-NUMERIC.
5. Test WORKFLD-NUMERIC for valid numeric value.
6. Assign FIELD-EDIT-ERROR, based on outcome of decimal point placement test and numeric validation.

IDATE

Function

Checks input for valid date format, *mm/dd/yy* or *mm-dd-yy* (if *TPI-datefield-LTH* is 8) or *mmddyy* (if *TPI-datefield-LTH* is 6), and stores it as *yymmdd*.

Note: If the INTDATE parameter in the macro TLNIIS is set to 'I,' a call to IINTLDT is generated in place of a call to IDATE. For more information, see IINTLDT.

Parameters

Type	Name	Description
Input	<i>TPI-datefield-LTH</i>	Length of the input field. It must be a halfword binary with a value of 6 or 8.
Input	<i>TPI-datefield</i>	Contents of the TP input buffer (the date to be edited).
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none"> ■ Spaces— No error was detected. ■ G123—An error was detected during processing of an item in <i>mmddyy</i> or <i>ddmmyy</i> format. ■ G130— An error occurred during processing of an item in <i>mm/dd/yy</i>, <i>dd/mm/yy</i>, <i>mm-dd-yy</i>, or <i>dd-mm-yy</i> format.
Output	WORKFLD-NUMERIC	Result of the edit in a COBOL PIC S9(11)V9(7) field, the date is laid out as 00000yymmdd0000000. For PL/I PIC (10)9V(4)9T field, the date is laid out as or 0000yymmdd00000. The program moves this to the appropriate storage area after return from the edit routine.

Edit processing

1. If the field length is eight, check for valid delimiters. If invalid delimiters are found, exit with error code G130.
2. Move date to WORKFLD-NUMERIC and test for valid month, day, and year. If invalid, return appropriate error code. Otherwise, return edited date in WORKFLD-NUMERIC.

Note: *Yy* is tested for only valid numeric characters. *Dd* and *mm* are edited for valid characters.

IDOLLAR**Function**

Checks input for valid dollar format, with a period (.) separating dollars and cents.

Note: If the INTDATE parameter in the macro TLNIIS is set to 'I', a call to ICURRNCY is generated in place of a call to IDOLLAR. For more information, see ICURRNCY.

Parameters

Type	Name	Description
Input	<i>TPI-fieldname</i> -LTH	Length of the input field.
Input	<i>TPI-fieldname</i>	Contents of the TP input buffer (the dollar amount to be edited).
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none">■ Spaces— No error was detected■ G121— An error was detected during the edit
Output	WORKFLD-NUMERIC	Result of the edit in the COBOL PIC S9(11)V9(7) format or PL/I PIC (10)9V(4)9T format.

Edit processing

1. Strip incoming TPI field of leading and trailing blanks.
2. Test the third from the last position for a valid decimal point (when INTDATE=U) or comma (when INTDATE=I).
3. Strip the decimal point or comma and move the justified number to WORKFLD-NUMERIC.
4. Zero-fill the value in WORKFLD-NUMERIC.
5. Test WORKFLD-NUMERIC for a valid numeric value.

IFLNULL**Function**

Checks a field for a valid number. This allows for a leading sign and an optional decimal point.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input field.
Input	TPI- <i>fieldname</i>	Contents of the TP input buffer (the number to be edited).
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none"> ■ Spaces— No error was detected ■ NXXX— An error was detected during the edit
Output	WORKFLD-NUMERIC	Result of the edit in the COBOL S9(11)V9(7) format or PL/I PIC (10)9V(4)9T format.
Output	WK- <i>fieldname</i> -NN	Null indicator variable.

Edit processing

If the input field is blank, set the null indicator to -1. Otherwise, set the null indicator to 0 and:

1. Skip leading blanks.
2. Check to see if the next character is a plus (+) or minus (-) sign.
3. Move all remaining characters, up to the decimal point, or a space, to a temporary work area DIGIT-GROUP.
4. If DIGIT-GROUP has more than 11 digits in a COBOL program or more than 10 digits in a PL/I program, exit with error code NXXX. Otherwise, move the contents of DIGIT-GROUP to a second temporary work area WS-FIELD-NUM.
5. If a decimal was found, place all numeric characters following the decimal point into DIGIT-GROUP.
6. If the resulting DIGIT-GROUP has more than seven digits in a COBOL program or more than five digits in a PL/I program, exit with error code NXXX. Otherwise, move DIGIT-GROUP to WS-FIELD-NUM following the original digits and a decimal point.
7. If the remaining unprocessed characters are blank, perform the following calculation to obtain the correct sign:
$$\text{WORKFLD-NUMERIC} = 0 + | - \text{WS-FIELD-NUM}$$

+ | - is the sign stored as the first nonblank character from the input field. If no character is found, plus (+) is assumed.
8. If any of the remaining unprocessed characters are not blanks, exit with error code NXXX.

IFLOAT**Function**

Checks a field for a valid number. This allows for a leading sign and optional decimal point.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input field.
Input	TPI- <i>fieldname</i>	Contents of the TP input buffer (the number to be edited).
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are:

Type	Name	Description
		<ul style="list-style-type: none"> ■ Spaces— No error was detected ■ G122— An error was detected during the edit
Output	WORKFLD-NUMERIC	Result of the edit in the COBOL S9(11)V9(7) format or PL/I PIC (10)9V(4)9T format.

Edit processing

1. Skip leading blanks.
2. Check to see if the next character is a plus (+) or minus (-) sign.
3. Move all remaining characters, up to the decimal point, or a space, in a temporary work area DIGIT-GROUP.
4. If DIGIT-GROUP has more than eleven digits in a COBOL program or more than nine digits in a PL/I program, exit with error code G122. Otherwise, move the contents of DIGIT-GROUP to a second temporary work area WS-FIELD-NUM.
5. If a decimal was found, place all the following numeric characters in DIGIT-GROUP.
6. If there are more than seven digits in the resulting DIGIT-GROUP value, exit with error code G122. Otherwise, move DIGIT-GROUP to WS-FIELD-NUM following the original digits and a decimal point.
7. If the remaining unprocessed characters are blank, perform the following calculation to obtain the correct sign:

$$\text{WORKFLD-NUMERIC} = 0 \{+|- \} \text{ES-FIELD-NUM.}$$

+|- is the sign stored as the first nonblank character from the input field. If no character is found, plus (+) is assumed.

If any of the remaining unprocessed characters are not blanks, exit with error code G122.

IFULLCAR

Function

Verify that there are no blanks in an input field. The difference between IFULLCAR and INBALPHA is that IFULLCAR allows no blanks at all. INBALPHA allows leading and trailing blanks.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input field.
Input	TPI- <i>fieldname</i>	Contents of the TP input buffer (the character field being checked for blanks).
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none">■ Spaces— No error was detected■ G122— An error was detected during the edit
Output	WORKFLD-ALPHA	Edited field. It is a 256-byte alphanumeric field.

Edit processing

1. Scan for blanks.
2. Return an error code of G122 if any blanks are found. Otherwise, move the field to WORKFLD-ALPHA and exit.

IFORMAT

Function

Reformats input.

Specification for this special edit is made by entering the FORMAT command in the SPEC field on the Update Output/Input/Outin Field screen.

See Update Output/Input/Outin Field or Update Batch Output Fields for more information.

IFULLNUM

Verify that all input characters in the field are numeric.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input field.

Type	Name	Description
Input	<i>TPI-fieldname</i>	Contents of the TP input buffer (the number being checked).
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none"> ■ Spaces— No error was detected ■ G124— An error was detected during the edit
Output	WORKFLD-NUMERIC	Edited field in the COBOL S9(11)V9(7) format or the PL/I PIC (10)9V(4)9T format.

Edit processing

1. Test all positions for valid numeric characters.
2. If any non-numeric characters are found, exit with error code G124.
3. Right-justify and zero-fill the input value and move it to WORKFLD-NUMERIC.

IHEX

Function

Converts a hex-character string to the hexadecimal equivalent for each pair of hex-character representations (for example, 'C1F2' converts to 'A1').

Parameters

Type	Name	Description
Input	<i>TPI-fieldname</i> -LTH	Length of the edited field. This must be between 1 and 256.
Input	<i>TPI-fieldname</i>	Field to be converted by this routine, from the TP input buffer.
Output	FIELD-EDIT-ERROR	Error status of the edit.
Output	WORKFLD-ALPHA	Converted value.

Edit processing

1. Determine the length of the input field.
2. Use the assembler language TRANSLATE instruction to convert the output from hex-representation to its character equivalent.
3. Assign FIELD-EDIT-ERROR.

IHEXA**Function**

Converts a hex-character string of one to eight bytes to the hexadecimal equivalent for each pair of hex-character representations (for example, 'C1F2' converts to 'A1'). This field edit is designed specifically for converting addresses.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the edited field. This must be between 1 and 256.
Input	TPI- <i>fieldname</i>	Field to be converted by this routine, from the TP input buffer.
Output	FIELD-EDIT-ERROR	Error status of the edit.
Output	WORKFLD-ALPHA	Converted value.

Edit processing

1. Determine the length of the input field.
2. Use the assembly language TRANSLATE instruction to convert the output from hex-representation to hexadecimal.
3. Right-justify the input into the full-word output field.
4. Assign FIELD-EDIT-ERROR.

IINTCDT

Function

Validates a date entered as *dd/mm/ccyy* or *dd-mm-ccyy* (if *TPI-datefield-LTH* is 10) and stores it in *ccyyymmdd* format.

A call to this field edit is generated in either of these cases:

- It is requested directly (that is, 'IINTCDT' is identified as the input field edit)
- The INTDATE parameter in TLNIIS is set to 'I' and CDATE is identified as the input field edit

Parameters

Type	Name	Description
Input	<i>TPI-datefield-LTH</i>	Length of the input field. It must be eight or ten bytes.
Input	<i>TPI-datefield</i>	Field to be validated by this routine, from the TP input buffer.
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none"> ■ Spaces— No error was detected ■ ERR1— Length error ■ G123— An error was detected during validation of an eight-byte date ■ G130— An error was detected during validation of a ten-byte date
Output	WORKFLD-NUMERIC	Converted date. In a COBOL PIC S9 (11) V9 (7) field, the date is laid out as 000ccyyymmdd00000000. In a PL/I PIC (10)9V (4)9T field, the date is laid out as 00ccyyymmdd000000.

Edit processing

1. Determine the input field length.
2. If 10, check for valid delimiters ('/' or '-'). If invalid delimiters are found, set FIELD-EDIT-ERROR to G130.
3. Move date to WORKFLD-NUMERIC and test for valid day, month, century, and year.
4. If invalid, set FIELD-EDIT-ERROR to the appropriate error code. Otherwise, return edited date in WORKFLD-NUMERIC.

Note: *Cc* and *yy* are tested only for valid numeric characters. *Dd* and *mm* are edited for both numeric and valid month-day combinations.

IINTCJUL**Function**

Validates a date entered as *dd/mm/ccyy* or *dd-mm-ccyy* (if *TPI-datefield-LTH* is 10) and stores it in *ccyyddd* format.

A call to this field edit is generated in either of these cases:

- It is requested directly (that is, 'INTCJUL' is identified as the input field edit)
- The INTDATE parameter in TLNIIS is set to 'I' and CJULIAN is identified as the input field edit

Parameters

Type	Name	Description
Input	<i>TPI-datefield-LTH</i>	Length of the input field. It must be eight or ten bytes.
Input	<i>TPI-datefield</i>	Field to be validated by this routine, from the TP input buffer.
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none">■ Spaces— No error was detected■ ERR1— Length error
Output	WORKFLD-NUMERIC	Converted date. In a COBOL PIC S9(11)V9(7) field, the date is laid out as 0000ccyyddd0000000. In a PL/I PIC (10)9V(4)9T field, the date is laid out as 000ccyyddd000000.

Edit processing

1. Determine the input field length.
2. If 10, check for valid delimiters ('/' or '-'). If invalid delimiters are found, set FIELD-EDIT-ERROR to ERR1.
3. Test for valid day, month, and year. If invalid, set FIELD-EDIT-ERROR to the ERR1 error code. Otherwise, convert the edited date to *ccyyddd* and move it to WORKFLD-NUMERIC.

Note: *Cc* and *yy* are tested only for valid numeric characters. *Dd* and *mm* are edited for both numeric and valid month-day combinations.

ICJULIAN**Function**

Checks input for a valid date format, *mm/dd/ccyy* or *mmd-dd-ccyy* (if TPI-*datefield*-LTH is 10) or *mmddccyy* (if TPI-*datefield*-LTH is 8).

Note: If the INTDATE parameter in TLNIIS is set to 'I,' a call to IINTCJUL is generated in place of a call to ICJULIAN. See IINTCJUL later in this appendix.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input field. It must be eight or ten bytes.
Input	TPI- <i>fieldname</i>	Field to be validated by this routine, from the TP input buffer (the date to be edited).
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none"> ■ Spaces— No error was detected ■ ERR1— Length error
Output	WORKFLD-NUMERIC	Julian date in a COBOL PIC S9(11)V9(7) format laid out as 0000 <i>ccyyddd</i> 0000000 or a PL/I PIC (10)9V(4)9T format laid out as 000 <i>ccyyddd</i> 00000 in PL/I.

Edit processing

1. If the length is 10, test for valid delimiters ('/' or '-'). Exit with error code ERR1 if they are not found.
2. Test for valid day, month, and year (cc is checked simply for valid numeric value). Exit with error code ERR1 if anyone is invalid.
3. Convert the date to Julian format (ccyyddd) and move the results to WORKFLD-NUMERIC.

IINTLDT**Function**

Validates a date entered as *dd/mm/yy* or *dd-mm-yy* (if *TPI-datefield-LTH* is 8), or *ddmmyy* (if *TPI-datefield-LTH* is 6), and stores it in *yymmdd* format

A call to this field edit is generated in either of these cases:

- It is requested directly (that is, 'IINTLDT' is identified as the input field edit)
- The INTDATE parameter in TLNIIS is set to 'I' and DATE is identified as the input field edit

Parameters

Type	Name	Description
Input	<i>TPI-datefield-LTH</i>	Length of the input field. It must be six or eight bytes.
Input	<i>TPI-datefield</i>	Field to be validated by this routine, from the TP input buffer.
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none">■ Spaces— No error was detected■ G123— An error was detected during validation of a six-byte date■ G130— An error was detected during validation of an eight-byte date

Type	Name	Description
Output	WORKFLD-NUMERIC	Converted date. In a COBOL PIC S9(11)V9(7) field, the date is laid out as 00000yyymmdd0000000. In a PL/I PIC (10)9V(4)9T field, the date is laid out as 0000yyymmdd00000.

Edit processing

1. Determine the input field length.
2. If 8, check for valid delimiters ('/' or '-'). If invalid delimiters are found, set FIELD-EDIT-ERROR to G130.
3. Move date to WORKFLD-NUMERIC and test for valid day, month, and year.
4. If invalid, set FIELD-EDIT-ERROR to the appropriate error code. Otherwise, return edited date in WORKFLD-NUMERIC.

Note: Yy is tested only for valid numeric characters. Dd and mm are edited for both numeric and valid month-day combinations.

IINTLJUL

Function

Validates a date entered as *dd/mm/yy* or *dd-mm-yy* (if TPI-datefield-LTH is 8) or *ddmmyy* (if TPI-datefield-LTH is 6), and store it in *yyddd* format.

A call to this field edit is generated in either of these cases:

- It is requested directly (that is, 'IINTLJUL' is identified as the input field edit)
- The INTDATE parameter in TLNIIS is set to 'I' and JULIAN is identified as the input field edit

Parameters

Type	Name	Description
Input	TPI-datefield-LTH	Length of the input field. It must be six or eight bytes.
Input	TPI-datefield	Field to be validated by this routine, from the TP input buffer.
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none"> ■ Spaces— No error was detected.

Type	Name	Description
		<ul style="list-style-type: none"> ■ ERR1— An error was detected during validation.
Output	WORKFLD-NUMERIC	Converted date. In a COBOL PIC S9(11)V9(7) field, the date is laid out as 000000yyddd0000000. In a PL/I PIC (10)9V(4)9T field, the date is laid out as 00000yyddd000000.

Edit processing

1. Determine the input field length.
2. If 8, check for valid delimiters ('/' or '-'). If invalid delimiters are found, set FIELD-EDIT-ERROR to ERR1.
3. Test for valid day, month, and year. If invalid, set FIELD-EDIT-ERROR to ERR1. Otherwise, convert edited date to *yyddd* and move it to WORKFLD-NUMERIC.

Note: Yy is tested only for valid numeric characters. Dd and mm are edited for both numeric and valid month-day combinations.

IJULIAN

Function

Checks input for a valid date format, *mm/dd/yy* or *mm-dd-yy* (if TPI-*datefield*-LTH is 8) or *mmddy* (if TPI-*datefield*-LTH is 6).

Note: If the INTDATE parameter in the macro TLNIIS is set to 'I', a call to IINTLJUL is generated in place of a call to IJULIAN. For more information, see IINTLJUL.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input field. This must be 6 or 8.
Input	TPI- <i>fieldname</i>	Contents of the TP input buffer (the date to be edited).
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none"> ■ Spaces— No error was detected.

Type	Name	Description
		<ul style="list-style-type: none"> ■ ERR1— An error was detected during validation. ■
Output	WORKFLD-NUMERIC	Returns the Julian date in a COBOL PIC S9(11)V9(7) field, the date is laid out as 000000yyddd0000000 in a PL/I PIC (10)9V(4)9T field, the date is laid out as 00000yyddd00000.

Edit processing

1. If the field length is 8, test for valid delimiters (/ or -). Exit with error code ERR1 if they are not found.
2. Test for valid day, month, and year. (Yy is checked simply for valid numeric value.) Exit with error code ERR1 if any one is invalid.
3. Convert the date to Julian format of yyddd and move the results to WORKFLD-NUMERIC.

ILALPHA**Function**

Left-justifies a character field.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input fields.
Input	TPI- <i>fieldname</i>	Contents of the TP input buffer (the field to be left-justified).
Output	WORKFLD-ALPHA	Edited field as a 256-byte alphanumeric field.

Edit processing

1. Start on left and search for the first nonblank character.
2. Compute length of remaining field.
3. Use the result to left-justify the result in WORKFLD-ALPHA.

ILNULL

Function

Left-justifies a character field.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input field.
Input	TPI- <i>fieldname</i>	Contents of the TP input buffer (the field to be left-justified).
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: ■ Spaces— No error was detected
Output	WORKFLD-ALPHA	Result of the edit as a 256-byte alphanumeric field.
Output	WK- <i>fieldname</i> -NN	Null indicator variable.

Edit processing

If the input field is blank, set the null indicator to -1. Otherwise, set the null indicator to 0 and:

1. Start on the left and search for the first nonblank character.
2. Compute length of remaining field.
3. Use the result to left-justify the remaining characters into WORKFLD-ALPHA.

ILVCHAR

Function

Left-justifies a variable-length character field.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input fields.
Input	TPI- <i>fieldname</i>	Contents of the TP input buffer (the variable length field to be left-justified).

Type	Name	Description
Output	WORKFLD-VCHAR	Edited field as a 256-byte alphanumeric variable-length field.

Edit processing

1. Start on left and search for the first nonblank variable-length character.
2. Search for the last nonblank variable-length character.
3. Compute length of remaining field.
4. Use the result to left-justify the result in WORKFLD-VCHAR.
5. Move the length to the length part of WORKFLD-VCHAR.

ILVNULL

Function

Left-justifies a variable-length character field.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input field.
Input	TPI- <i>fieldname</i>	Contents of the TP input buffer (the variable-length field to be left-justified).
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none"> ■ Spaces— No error was detected
Output	WORKFLD-VCHAR	Result of the edit as a 256-byte variable-length alphanumeric field.
Output	WK- <i>fieldname</i> -NN	Null indicator variable.

Edit processing

If the input field is blank, set the null indicator to -1. Otherwise, set the null indicator to 0, and:

1. Start on the left and search for the first nonblank character.
2. Search for the last nonblank character.
3. Compute length of remaining field.
4. Use the result to left-justify the remaining characters into WORKFLD-VCHAR.
5. Move the length of the remaining field to the length portion of the WORKFLD-VCHAR.

INBALPHA**Function**

Checks for imbedded blanks. INBALPHA differs from IFULLCAR because it allows leading and trailing blanks. IFULLCAR allows neither.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input field.
Input	TPI- <i>fieldname</i>	Contents of the TP input buffer (the character field being checked for imbedded blanks).
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none">■ Spaces— No error was detected■ G122— An error was detected during the edit
Output	WORKFLD-ALPHA	Edited field in a 256-byte alphanumeric field.

Edit processing

1. Search for the first nonblank character.
2. Search for the next nonblank character or end-of-field.
3. If end-of-field is reached before the next nonblank, move the input field to WORKFLD-ALPHA and exit without an error code.
4. If a blank is found, make sure all remaining characters are blank. If they are, move the input field to WORKFLD-ALPHA and exit without an error code. If another nonblank is found, exit with error code G122.

INBNUL**Function**

Checks for imbedded blanks. This edit allows for leading and trailing blanks.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input field.
Input	TPI- <i>fieldname</i>	Contents of the TP input buffer (the field being checked for imbedded blanks).
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none">■ Spaces— No error was detected■ N122— An error was detected during the edit
Output	WORKFLD-ALPHA	Result of the edit as a 256-byte alphanumeric field.
Output	WK- <i>fieldname</i> -NN	Null indicator variable.

Edit processing

If the input field is blank, set the null indicator to -1. Otherwise, set the null indicator to 0, and:

1. Search for the first nonblank character.
2. Search for the next nonblank character or end-of-field.
3. If end-of-field is reached before the next nonblank character, move the input field to WORKFLD-ALPHA and exit without an error code.
4. If a blank is found, make sure all remaining characters are blank. If they are, move the input field to WORKFLD-ALPHA and exit without an error code. If another nonblank character is found, exit with error code N122.

INBVCHAR**Function**

Checks for imbedded blanks. This edit allows leading and trailing blanks.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input fields.

Type	Name	Description
Input	<i>TPI-fieldname</i>	Contents of the TP input buffer (the variable-length character field being checked for imbedded blanks).
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none"> ■ Spaces— No error was detected ■ G122— An error was detected during the edit
Output	WORKFLD-VCHAR	Edited field in a 256-byte alphanumeric variable-length field.

Edit processing

1. Search for the first nonblank character.
2. Search for the next nonblank character or end-of-field.
3. If end-of-field is reached before the next nonblank, move the input field to WORKFLD-VCHAR and exit without an error code.
4. If a blank is found, make sure all remaining variable-length characters are blank. If they are, move the input field to the character part of WORKFLD-VCHAR, move the length to the length part of WORKFLD-VCHAR and exit without an error code. If another nonblank is found, exit with error code G122.

INBVNULL

Function

Checks for imbedded blanks in a variable-length character field. This edit allows for leading and trailing blanks.

Parameters

Type	Name	Description
Input	<i>TPI-fieldname-LTH</i>	Length of the input field.
Input	<i>TPI-fieldname</i>	Contents of the TP input buffer (the variable-length field being checked for imbedded blanks).

Type	Name	Description
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none"> ■ Spaces— No error was detected ■ N122— An error was detected during the edit
Output	WORKFLD-VCHAR	Result of the edit as a 256-byte variable-length alphanumeric field.
Output	WK- <i>fieldname</i> -NN	Null indicator variable.

Edit processing

If the input field is blank, set the null indicator to -1. Otherwise, set the null indicator to 0, and:

1. Search for the first nonblank character.
2. Search for the next nonblank character or end-of-field.
3. If end-of-field is reached before the next nonblank character, move the input field to the character portion of WORKFLD-VCHAR, move the length to the length portion of WORKFLD-VCHAR, and exit without an error code.
4. If a blank is found, make sure all remaining variable-length characters are blank. If they are, move the input field to the character portion of the WORKFLD-VCHAR, move the length to the length portion of WORKFLD-VCHAR, and exit without an error code. If another nonblank character is found, exit with error code N122.

INULL

Function

Checks for a completely blank field.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input field.
Input	TPI- <i>fieldname</i>	Contents of the TP input buffer (the field to be checked).

Type	Name	Description
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none"> ■ Spaces— No error was detected
Output	WORKFLD-ALPHA	Returns the result of the edit as a 256-byte alphanumeric field.
Output	WK- <i>fieldname</i> -NN	Null indicator variable.

Edit processing

If the input field is blank, set the null indicator to -1. Otherwise, set the null indicator to 0, and move the contents of TPI-*fieldname* to WORKFLD-ALPHA.

INUMERIC

Function

Checks for a valid integer. This field edit allows for leading and trailing blanks, but not imbedded blanks and non-numeric characters.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input field.
Input	TPI- <i>fieldname</i>	Contents of the TP input buffer (the number to be edited).
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none"> ■ Spaces— No error was detected ■ G122— An error was detected during the edit
Output	WORKFLD-NUMERIC	Edited field in the COBOL S9(11)V9(7) format or the PL/I (10)9V(4)9T format.

Edit processing

1. Search for the first nonblank character. Make sure it is numeric.
2. Search for the earlier of the next nonblank or the end-of-field. Make sure there are no imbedded blanks and that all characters are numeric.
3. If no imbedded blanks or non-numeric characters are found, right-justify and zero-fill the field in WORKFLD-NUMERIC and exit. Otherwise, exit with error code G122.

INUMNULL**Function**

Checks for a valid integer. This edit allows leading and trailing blanks, but not imbedded blanks and non-numeric characters.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input field.
Input	TPI- <i>fieldname</i>	Contents of the TP input buffer (the field being edited).
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none">■ Spaces— No error was detected■ N122— An error was detected during the edit
Output	WORKFLD-NUMERIC	Result of the edit in the COBOL S9(11)V9(7) format or PL/I (10)9V(4)9T format.
Output	WK- <i>fieldname</i> -NN	Null indicator variable.

Edit processing

If the input field is blank, set the null indicator to -1. Otherwise, set the null indicator to 0, and:

1. Search for the first nonblank character. Make sure it is numeric.
2. Search for the earlier of the next nonblank character or end-of-field. Make sure there are no imbedded blanks and that all characters are numeric.
3. If no imbedded blanks or non-numeric characters are found, right-justify and zero-fill the field in WORKFLD-NUMERIC, and exit the edit. Otherwise, exit the edit with an error code of N122.

ISTATE**Function**

Verify that a valid two-character postal state code is entered.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input field. It must always be 2.
Input	TPI- <i>fieldname</i>	Contents of the TP input buffer (the state code to be checked).
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none"> ■ Spaces— No error was detected ■ G133— An error was detected during the edit
Output	WORKFLD-ALPHA	Returns the edited field in a 256-byte alphanumeric field.

Edit processing

1. Move TPI-*fieldname* to WORKFLD-ALPHA.
2. Verify the contents of WORKFLD-ALPHA against a state-code table that includes the 50 states, the District of Columbia, and Puerto Rico.
3. If there is a match, exit with no error. Otherwise, exit with error code G133.

IVCHAR

Function

Moves a variable-length character field.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input field.
Input	TPI- <i>fieldname</i>	Contents of the TP input buffer (the variable-length character field to be moved).
Output	WORKFLD-VCHAR	Edited field in a 256-byte (maximum) alphanumeric variable-length character field and the length of the variable-length character field.

Edit processing

1. Search for the last nonblank character.
2. Determine remaining length of field.
3. Move the input field to the character part of WORKFLD-VCHAR.
4. Move the length to the length part of WORKFLD-VCHAR.

IVNULL

Function

Sets a null indicator based on the contents of a variable-length character field. This edit can be used to check for a blank input field.

Parameters

Type	Name	Description
Input	TPI- <i>fieldname</i> -LTH	Length of the input field.
Input	TPI- <i>fieldname</i>	Contents of the TP input buffer (the variable-length field to be checked).
Output	FIELD-EDIT-ERROR	Error status of the edit. Possible values are: <ul style="list-style-type: none">■ Spaces— No error was detected

Type	Name	Description
Output	WORKFLD-VCHAR	Result of the edit as a 256-byte variable-length alphanumeric field.
Output	WK- <i>fieldname</i> -NN	Null indicator variable.

Edit processing

If the input field is blank, set the null indicator to -1. Otherwise, set the null indicator to 0, and:

1. Search for the first nonblank character.
2. Determine the remaining length of field.
3. Move the input field to the character portion of WORKFLD-VCHAR.
4. Move the length to the length portion of WORKFLD-VCHAR.

Calls to Field Edit Modules

Automatic calls to input or output edits, whether CA Telon- or user-supplied, always pass parameters in the order shown in the examples below.

Output edit processing

The following code shows an automatic call to an output edit, the CA Telon-supplied DATE output edit routine. DAY is the screen field to receive the edited output.

COBOL

```
CALL 'ODATE' USING TPO-DAY
      TPO-DAY-LTH,
      WORKFLD-NUMERIC.
```

PL/I

```
CALL ODATE (TPO_DAY,
            TPO_DAY_LTH,
            WORKFLD_NUMERIC);
```

Input edit processing

The following code shows the automatic call to an input edit, the CA Telon-supplied DATE input edit routine. DAY is the screen field holding the input to be edited. The edited data is returned to WORKFLD-NUMERIC.

COBOL

```
CALL 'IDATE' USING FIELD-EDIT-ERROR TPI-DAY-LTH  
          TPI-DAY  
          WORKFLD-NUMERIC.
```

PL/I

```
CALL IDATE (FIELD_EDIT_ERROR,  
          TPI_DAY_LTH,  
          TPI_DAY,  
          WORKFLD_NUMERIC);
```

Appendix B: Screen Program Identifiers

As a quick reference, this appendix lists CA Telon screens alphabetically by screen program ID or title.

Each list associates the screen title with the identifier of the program that displays the screen.

Screen Program IDs

The following table lists CA Telon screen program identifiers and their screen titles in alphabetical order by program identifier.

Program ID	Screen Title
B100	Batch Program Definition Menu
B110	Create/Update Batch Definition
B114	Show/Purge Batch Definition
B168	Update MVS Batch Environment
B1M1	List Merge Key Groups
B1M2	Update Group 99 Merge Keys
B1MA	List Match Keys
B1S1	List User Sorts
B1S2	Update Sort
B210	Create/Update Stored Procedure
B214	Show/Purge Stored Definition
B268	Update Stored Env
B269	Show/Purge Stored Environment
B2P1	Update Stored Parameters
B2P2	Stored Proc Parm
D100	Data Administration Menu
D111	Create/Update DBD
D112	Update Segment Default Data
D114	Update Dataset Default Data

Program ID	Screen Title
D115	List Search Fields
D116	Update DLIDSCs For Segment
D117	Show DBDs
D118	Update SSA/Command for DL/I DB/Segment
D11J	CICS Journal Default Data
D11Q	CICS Queue Default Data
D120	Verify DLIDSC Count
D141	Create/Update SQL Tables
D142	Specify Tables Being Joined
D143	Update SQL Join— Join Columns
D144	Update SQL Join— Access Columns
D145	Add/Update SQL Join Alias
D147	Show/Purge TB
D151	Update SQL Table
D152	Update IDMS SQL Table
D157	Show/Purge CA-IDMS/CA-Datacom SQL Tables/TLNROWS
D211	Create/Update PSB, File Group
D215	Update Sensitive Seg
D216	Delete Sensitive TLNROWS
D217	Show PSB, File Group
D401	List Data Administration Info
D402	List SQL
D411	Catalog/Import DB2 Tables
F100	Telon Design Facility Main Menu
F105	User Profile Maintenance Menu
F110	Update PFKEYS
F112	Update Program Defn Defaults
F113	Update Environment Defn Defaults
F114	Update Session Controls
F116	Color Profile (PWS Only)

Program ID	Screen Title
F120	TDF Installation Menu
F121	List TDF Users
F130	Update Installation Restrict
F131	Update PWS Desktop Security
F140	List Application Headers (Install Screen)
F144	List Application Headers
F154	Update Stored Environ Defn Default
F401	List/Purge TDF WIP Records
F402	List Data Administration Info
F403	List/Purge TNTDXW Records
M100	Prototyping Facility Menu
M120	N/A (Prototyping Full Screen Display - No Title)
M151	View Presentation Store
M200	List Presentation Stores
M401	List Panel Definitions
N110	Create/Update Nonterm. Definition
N114	Show/Purge Nonterm. Definition
P100	Panel Definition Menu
P103	N/A (Panel Image Full Screen Editor - No Title)
P104	Line Edit Screen Editor (Panel Image 24x80)
P113	N/A (Panel Image 27x132)
P114	Line Edit (Panel Image 27x132)
P150	Update Literal Field Data
P151	Update Input Field Data
P152	Update Output Field Data
P153	Update Outin Field Data
P154	Update Select Field Data
P155	Update Panel Fields
P156	Update HELPMMSG Parm
P157	Update Select Parms

Program ID	Screen Title
P158	Update Attribute Parm
P159	Update Mapout Parm
P161	List SRC, XFEDIT, SEGEDIT
P165	Update XFEDIT
P168	Update SEGEDIT
P170	Update Table Segloop
P175	Update File Segloop
P180	Update Literal Field
P181	Update Output/Input/Outin Field
P182	Update Select Field
P186	Update Parm List Extension
P255	Update Panel Fields
P280	Update Literal Field
P281	Update Output Field
P290	Update Panel Group
P401	List Definitions
PHL2	Telon Help Facility
PHLP	Telon Help Facility
PHLU	Edit ---- (Update Help Message)
S100	Online Program Definition Menu
S110	Create/Update Screen Definition
S112	Update Screen Parm
S114	Show/Purge Screen Definition
S125	Update Data Group
S127	Update DBMS Characteristics
S135	Update DBD
S136	Update Data Record
S137	Select New Row Name
S13J	Update CICS Journal Record
S13Q	Update CICS Queue Record

Program ID	Screen Title
S144	Field Extension
S145	Update Detail Data Access
S146	N/A (Update VSAM/Sequential Data Access Request
S147	N/A (Update SQL Data Access Request)
S149	Update TPPCB Data Access
S14J	N/A (Update Journal Data Access Request)
S14Q	N/A (Update CICS Queue Data Access Request)
S151	Edit ---- (Update Custom Code)
S159	List Custom Code
S161	Show/Purge TSO/IMS Screen Env
S162	Update TSO/IMS Screen Env
S163	Update IMS MFS
S164	Create/Update/Purge PL/I Execution Opts
S165	Update CICS Environment
S166	Show CICS Environment
S167	Create/Update IMS/DC Driver Env
S168	Create/Update IMS/DC Report Env
S184	Update User-Defined Datatypes
S185	Update DLI User I/O
S187	Select Columns
S210	Create/Update IMS/DC Driver Defn
S214	Show/Purge Driver Defn
S225	List Stored Procedures To Be Called
S240	Get Diagnostics List
S241	Get Diagnostics Statement (STMT)
S242	Get Diagnostics Statement (COND)
S243	Get Diagnostics Statement (COMB)
S244	Fetch Details
S310	Create/Update IMS/DC Report Defn
S314	Show/Purge IMS/DC Report Defn

Program ID	Screen Title
S401	List Environments
U100	Utilities Menu
Z100	Update Parameter Overflow
Z101	Show Parameter Overflow
Z102	Update Parameter Overflow

Screen Titles

The following table lists CA Telon screen program titles and their program identifiers in alphabetical order by screen title.

Screen Title	Program ID
Batch Program Definition Menu	B100
Catalog/Import DB2 Tables	D411
CICS Journal Default Data	D11J
CICS Queue Default Data	D11Q
Color Profile (PWS Only)	F116
Create/Update Batch Definition	B110
Update Data Group	S125
Create/Update IMS/DC Driver Defn	S210
Create/Update DBD	D111
Create/Update SQL Tables	D141
Update File Segloop	P175
Create/Update IMS/DC Report Defn	S310
Create/Update Nonterm. Definition	N110
Create/Update PSB, File Group	D211
Create/Update Screen Definition	S110
Update SSA/Command For DL/I DB/Segment	D118
Create/Update Stored Procedure	B210
Update Table Segloop	P170

Screen Title	Program ID
Data Administration Menu	D100
Edit ---- (Update Custom Code)	S151
N/A (Panel Image Full Screen Editor - No Title)	P103
Fetch Details	S244
Field Extension	S144
Get Diagnostics statement (COMB)	S243
Get Diagnostics Statement (COND)	S242
Get Diagnostics Statement (STMT)	S241
GET DIAGNOSTICS LIST	S240
Telon Help Facility	PHL2
Edit ---- (Update Help Message)	PHLU
Line Edit (Panel Image 27x132)	P114
Line Edit Screen Editor (Panel Image 24x80)	P104
List Application Headers (Install Screen)	F140
List SRC, XFEDIT, SEGEDIT	P161
List Data Administration Info	D401
List Environments	S401
List SQL	D402
List Merge Key Groups	B1M1
List Panel Definitions	M401
List Definitions	P401
List Presentation Stores	M200
List Search Fields	D115
List Select Columns	S187
List User Sorts	B1S1
List Stored Procedures To Be Called	S225
List TDF Users	F121
List Custom Code	S159
Online Program Definition Menu	S100
Panel Definition Menu	P100

Screen Title	Program ID
Prototyping Facility Menu	M100
N/A (Panel Image 27x132)	P113
Select New Row Name	Z101
Show Parameter Overflow	S137
Show/Purge Batch Definition	B114
Show CICS Environment	S166
Show/Purge TB	D147
Show DBDs	D117
Show/Purge Driver Defn	S214
Show/Purge IMS/DC Report Defn	S314
Show/Purge Nonterm. Definition	N114
Show PSB, File Group	D217
Show/Purge Screen Definition	S114
Show/Purge CA-IDMS/CA-Datcom SQL Tables/TLNROWS	D157
Show/Purge Stored Definition	B214
Show/Purge Stored Environment	B269
Show/Purge TSO/IMS Screen Env	S161
Specify Tables Being Joined	D142
Field Extension	S144
Stored Proc Parm	B2P2
Update Input Field Data	P151
Update Literal Field Data	P150
Update Outin Field Data	P153
Update Output Field Data	P152
Update Select Field Data	P154
TDF Installation Menu	F120
Telon Design Facility Main Menu	F100
Telon Help Facility	PHLP
Update Attribute Parm	P158
Update Literal Field	P280

Screen Title	Program ID
Update Output Field	P281
Update XFEDIT	P165
N/A (Update Journal Data Access Request)	S14J
Update Dataset Default Data	D114
Update IDMS SQL Table	D152
Update MVS Batch Environment	B168
Update SQL Table	D151
Update CICS Journal Record	S13J
N/A (Update CICS Queue Data Access Request)	S14Q
Update CICS Queue Record	S13Q
Update CICS Environment	S165
Update Data Record	S136
Update DBD	S135
Update DLI User I/O	S185
Update Segment Default Data	D112
Update DBMS Characteristics	S127
Update Detail Data Access	S145
Update DLIDSCs For Segment	D116
Update Environment Defn Defaults	F113
Update Group 99 Merge Keys	B1M2
Update HELPMMSG Parm	P156
Update Installation Restrict	F130
Update PWS Desktop Security	F131
Update IDMS Record Information	S138
Update IDMS Set/Area Information	S139
List Application Headers	F144
Update IMS MFS	S163
Create/Update IMS/DC Driver Env	S167
Create/Update IMS/DC Report Env	S168
Update Literal Field	P180

Screen Title	Program ID
Update Mapout Parm	P159
List Match Keys	B1MA
Update Output/Input/Outin Field	P181
Update Panel Fields	P255
Update Panel Fields	P155
Update Panel Group	P290
Update Parm List Extension	P186
Update Parameter Overflow	Z102
Update Parameter Overflow	Z100
Update PFKEYS	F110
Create/Update/Purge PL/I Execution Opts	S164
Update Program Defn Defaults	F112
Update SEGEDIT	P168
Update Select Field	P182
Update Select Parm	P157
Update Sensitive IDMS Segments	D218
Update Sensitive Seg	D215
Delete Sensitive TLNROWS	D216
Update Session Controls	F114
Update Sort	B1S2
N/A (Update SQL Data Access Request)	S147
Update SQL Join— Access Columns	D144
Add/Update SQL Join Alias	D145
Update SQL Join— Join Columns	D143
Update Stored Env	B268
Update Stored Environ Defn Default	F154
Update Stored Parameters	B2P1
Update TPPCB Data Access	S149
Update User-Defined Datatypes	S184
Update TSO/IMS Screen Env	S162

Screen Title	Program ID
N/A (Update VSAM/Sequential Data Access Request	S146
Update Screen Parm	S112
User Profile Maintenance Menu	F105
Utilities Menu	U100
Verify DLIDSC Count	D120
N/A (Prototyping Full Screen Display - No Title)	M120
View Presentation Store	M151
List Data Administration Info	F402
List/Purge TDF WIP Records	F401
List/Purge TNTDXW Records	F403

Index

A

- ^N (NOTNULL) • 132
 - Create/Update IDMS SQL Tables/TLNROWS • 132

A

- ABC field • 96, 277
 - Create/Update PSB Or File Group • 96
 - Update DBMS Characteristics • 277
- AC • 132, 150
 - Add/Update SQL Join Alias • 150
 - Create/Update IDMS SQL Tables/TLNROWS • 132
- AC field • 124, 147, 332
 - Create/Update SQL Tables/TLNROWS • 124
 - Select Columns • 332
 - Update SQL Join? Access Columns • 147
- ACCESS COLUMNS field • 145
 - Update SQL Join? Join Columns • 145
- ACCESS field • 73, 106, 124, 147, 291
 - Create/Update DBD • 73
 - Create/Update SQL Tables/TLNROWS • 124
 - Show/Update Data Set Default Data • 106
 - Update Data Set Record • 291
 - Update SQL Join? Access Columns • 147
- ACCMODE field • 106, 291
 - Show/Update Data Set Default Data • 106
 - Update Data Set Record • 291
- ALARM field • 30, 394
 - Update Program Definition Defaults • 30
- ALIAS • 132, 141, 150
 - Add/Update SQL Join Alias • 150
 - Create/Update IDMS SQL Tables/TLNROWS • 132
 - Update SQL ALIAS • 141
- ALIAS field • 124, 147, 332
 - Create/Update SQL Tables/TLNROWS • 124
 - Select Columns • 332
 - Update SQL Join? Access Columns • 147
 - use with COPY • 124
 - use with FROM • 147
- APPLID field • 30, 394, 401, 416, 429, 462
 - Update Program Definition Defaults • 30
- ASUTIME • 47
 - ASUTIME • 47

- ATTRINT field • 180, 185, 189, 199
 - Update Attribute Parameter • 180
 - Update Literal Fields • 185
 - Update Output/Input/Outin Field • 189
 - Update Select Fields • 199
- ATTRPRO field • 180, 189, 199
 - Update Attribute Parameter • 180
 - Update Output/Input/Outin Field • 189
 - Update Select Fields • 199
- auto exec • 199
 - create/update segments requested for • 199
- auto exec requests • 267
 - online programs • 267
- AUTOEXEC requests • 267, 429
 - defined • 267
 - match master (MATCHM) • 267
 - match transaction (MATCHT) • 267
- AUX/MAIN field • 113, 348, 356
 - Update CICS Queue Data Access • 356
 - Update CICS Queue Default Data • 113
 - Update CICS Queue Record • 348
- auxiliary storage • 113, 348, 356
 - queues • 113, 348, 356

B

- BASE DEFN field • 246, 458
 - Online Program Definition menu • 246
- BASE field • 65
 - Data Administration menu • 65
- batch • 442
 - panel definition example • 442
- batch definition • 25
 - TDF Main menu • 25
- BLANK.WHEN.SAME field • 446, 448
 - Update Batch Literal Fields • 446
 - Update Batch Output Fields • 448
- BLKSIZE field • 106
 - Show/Update Data Set Default Data • 106
- BLKSZ field • 291
 - Update Data Set Record • 291
- BMS • 170, 185, 189, 199, 534
 - control blocks, generation of • 170, 185, 189, 199
 - creation • 534
- BMS field • 37
 - Update Environment Definition Defaults • 37

BOOLOOP field • 90
 Create/Update SSA/Command For DL/I
 DB/Segment • 90

C

CALC field • 448
 Update Batch Output Fields • 448

CALL FUNC field • 215
 Update Segment Edit (SEGEDIT) • 215

CAPS field • 30, 394
 Update Program Definition Defaults • 30

CASE field • 156
 Panel Definition menu • 156
 procedure for modifying • 156

C-CNT field • 104, 281
 Select New Row Name • 281
 Update Sensitive TLNROWS • 104

CICS • 37, 55, 106, 343, 356, 361, 387, 534
 BMS • 37
 BMS map • 534
 DATA SET options • 106, 343
 environment • 55
 environment fields • 37
 handle conditions • 356, 361

CICS, program sections • 37
 PSBSCHD field • 37

CINCRE field • 228, 233
 Create/Update File SEGLOOP • 233
 Create/Update Table SEGLOOP • 228

CJNAME field • 117, 352
 Update CICS Journal Default Data • 117
 Update CICS Journal Record • 352

CMDCODE field • 90, 215
 Create/Update SSA/Command For DL/I
 DB/Segment • 90
 Update Segment Edit (SEGEDIT) • 215

CMND field • 77, 84, 285, 335
 Update Database Segment • 285
 Update DBD Segment • 77
 Update DL/I Detail Data Access • 335
 Update DLIDSCs For Segment Member • 84

CMPOPT field • 250, 401, 416, 429, 462
 Create/Update Screen Definition • 250

CNTGRP field • 448
 Update Batch Output Fields • 448

COBDIV field • 106, 291
 Show/Update Data Set Default Data • 106
 Update Data Set Record • 291

COBOL • 30

 specified as programming language • 30

COBVSky field • 106, 291
 Show/Update Data Set Default Data • 106
 Update Data Set Record • 291

COL field • 170, 176, 178, 180, 185, 189, 199, 426, 442, 446, 448
 Update Attribute Parameter • 180
 Update Batch Literal Fields • 446
 Update Batch Output Fields • 448
 Update Help Message Parameters • 176
 Update Literal Fields • 185
 Update Mapout Parameter • 178
 Update Output/Input/Outin Field • 189
 Update Panel Fields (Batch) • 442
 Update Panel Fields (Online) • 170
 Update Select Fields • 199

COLLID • 47
 COLLID • 47

color attribute • 30, 394
 Update Program Definition Defaults • 30

COLS field • 120
 Catalog/Import DB2 Tables • 120

COLSGLP field • 228, 233
 Create/Update File SEGLOOP • 233
 Create/Update Table SEGLOOP • 228

column • 124, 145, 147, 332
 accessing a • 124, 145, 147, 332
 naming a • 124, 145, 147

column 1 • 534
 ignore • 534
 printed output • 534

COLUMN field • 228, 233
 Create/Update File SEGLOOP • 233
 Create/Update Table SEGLOOP • 228

COLUMN NAME • 132, 141, 150
 Add/Update SQL Join Alias • 150
 Create/Update IDMS SQL Tables/TLNROWS • 132
 Update SQL ALIAS • 141

COLUMN NAME field • 124, 145, 147, 332
 Create/Update SQL Tables/TLNROWS • 124
 Select Columns • 332
 Update SQL Join? Access Columns • 147
 Update SQL Join? Join Columns • 145

column spacing • 228, 233
 Create/Update File SEGLOOP • 233
 Create/Update Table SEGLOOP • 228

columns • 120, 298, 416

- generated I/O sensitive, see SENCOLS field • 298
- GROUPBY clause • 298
- HAVING clause • 298
- keys for TELON generated I/O, see KY or KEYCOLS field • 298
- number of • 120
- ORDERBY clause, see ORDERBY field • 298
- COLUMNS DEFINED field • 104, 281
 - Select New Row Name • 281
 - Update Sensitive TLNROWS • 104
- Comma • 17
 - repeat symbol, use in • 17
- COMMAND • 132, 141, 150
 - Add/Update SQL Join Alias • 150
 - Create/Update IDMS SQL Tables/TLNROWS • 132
 - Update SQL ALIAS • 141
- Command Code field • 77, 84, 90, 335
 - overriding of • 77, 84
 - overriding the • 90, 335
- command line • 168
 - line edit mode • 168
- COMMAND POS field • 514
 - Prototyping Facility menu • 514
- COMMIT field • 106
 - Show/Update Data Set Default Data • 106
- COMPAT field • 96
 - Create/Update PSB Or File Group • 96
- COMPRESS STATEMENTS field • 55
 - Update Session Controls • 55
- COMPRESSED field • 534
 - Utilities menu • 534
- COMRETN • 47
 - COMRETN • 47
- concatenated key • 90, 277, 335
 - length of • 277
 - qualify an I/O request • 90
 - to qualify an I/O request • 335
- CONCATK field • 90, 335
 - Create/Update SSA/Command For DL/I DB/Segment • 90
 - Update DL/I Detail Data Access • 335
- CONSIS field • 156, 250
 - Create/Update Screen Definition • 250
 - Panel Definition menu • 156
- consistency edits • 156, 209, 215, 222, 228, 233
 - CONSIS field • 156

- Create/Update File SEGLOOP • 233
- Create/Update Table SEGLOOP • 228
- example • 222
- List Consistency Edits • 209
- Update Segment Edit (SEGEDIT) • 215
- CONTROL • 442, 454
 - GPTYPE field • 442, 454
- CONVERS field • 40, 410
 - Update Environment Definition Defaults • 40
- CONVERT • 448
 - FIELD • 448
- COPY • 132, 141
 - Create/Update IDMS SQL Tables/TLNROWS • 132
 - Update SQL ALIAS • 141
- COPY EDIT BASE field • 212, 215
 - Update Cross-Field Edit (XFEDIT) • 212
 - Update Segment Edit (SEGEDIT) • 215
- COPY field • 77, 106, 113, 117, 124, 285, 291, 348, 352, 486
 - Create/Update SQL Tables/TLNROWS • 124
 - Show/Update Data Set Default Data • 106
 - Update CICS Journal Default Data • 117
 - Update CICS Journal Record • 352
 - Update CICS Queue Default Data • 113
 - Update CICS Queue Record • 348
 - Update Data Set Record • 291
 - Update Database Segment • 285
 - Update DBD Segment • 77
- COPY member • 77, 462
 - I/O area of segment, see COPYLV1 field • 77
 - segment definition, see COPY field • 77
- COPYLBL • 132, 141
 - Create/Update IDMS SQL Tables/TLNROWS • 132
 - Update SQL ALIAS • 141
- COPYLBL field • 77, 106, 113, 124, 285, 291, 348, 486
 - Create/Update SQL Tables/TLNROWS • 124
 - Show/Update Data Set Default Data • 106
 - Update CICS Queue Default Data • 113
 - Update CICS Queue Record • 348
 - Update Data Set Record • 291
 - Update Database Segment • 285
 - Update DBD Segment • 77
- COPYLV1 • 132, 141
 - Create/Update IDMS SQL Tables/TLNROWS • 132
 - Update SQL ALIAS • 141

COPYLV1 field • 77, 106, 113, 117, 124, 285, 291, 348, 352, 486
 Create/Update SQL Tables/TLNROWS • 124
 Show/Update Data Set Default Data • 106
 Update CICS Journal Default Data • 117
 Update CICS Journal Record • 352
 Update CICS Queue Default Data • 113
 Update CICS Queue Record • 348
 Update Data Set Record • 291
 Update Database Segment • 285
 Update DBD Segment • 77
CORRELATION NAME • 150
 Add/Update SQL Join Alias • 150
CORRELATION NAME field • 143, 145, 147
 Specify Tables Being Joined • 143
 Update SQL Join? Access Columns • 147
 Update SQL Join? Join Columns • 145
CPYCALL field • 298
 Update SQL Detail Data Access • 298
CPYINIT field • 298
 Update SQL Detail Data Access • 298
CPYKEY field • 298
 Update SQL Detail Data Access • 298
CPYTERM field • 298
 Update SQL Detail Data Access • 298
CQNAME field • 113, 348
 Update CICS Queue Default Data • 113
 Update CICS Queue Record • 348
cross-field edit • 212
 XFEDIT • 212
CURRENT field • 90, 335
 Create/Update SSA/Command For DL/I DB/Segment • 90
 Update DL/I Detail Data Access • 335
CURRENT ROW NAME field • 281
 Select New Row Name • 281
CURSCUS field • 250
 Create/Update Screen Definition • 250
CURSOR AT FIELD field • 212, 215
 Update Cross-Field Edit (XFEDIT) • 212
 Update Segment Edit (SEGEDIT) • 215
CURSOR field • 250
 Create/Update Screen Definition • 250
CUSTCODE field • 246, 458
 Online Program Definition menu • 246
custom code • 30, 228, 233, 246, 250, 298, 368, 371, 401, 416, 429, 462
 adding with PGMCUST • 30, 401, 416
 B-100-OUTPUT-EDITS • 250, 416

COLSGLP • 228, 233
CPYCALL field • 298
CPYINIT field • 298
CPYKEY field • 298
CPYTERM field • 298
creating • 250, 298, 401, 416
CURSCUS • 250
CUSTCODE field • 246
D-100-INPUT-INIT • 250
E-100-INPUT-EDITS • 250
following call (EXEC SQL) • 298
H-100-INPUT-TERM • 250
ICUST2 • 233
in the body of the call (EXEC SQL) • 298
OCUST1 • 228, 233
OCUST2 • 228, 233
OCUST3 • 228, 233
OINIT1 • 250
OINIT2 • 250, 416
PGMCUST field • 250
prior to the call (EXEC SQL) • 298
RESLTCC field • 298
RESLTPR field • 298
termination section • 250
X-100-CONSIG-EDITS • 250
custom code editor • 549, 551
 invoking • 549

D

data administration • 25
 TDF Main menu • 25
data group • 267
 separator line • 267
data set • 106, 291
 I/O • 106
 index • 106, 291
 label • 106, 291
 opening a • 106, 291
 record layout • 106, 291
DATA SET field • 106, 291, 493
 Show/Update Data Set Default Data • 106
 Update Data Set Record • 291
DB field • 122, 124
 SQL type • 122
DB2 • 25, 120, 143
 qualifier • 120, 143
 read-only access • 25
DBD • 65, 70, 73, 77, 81, 84, 90, 95, 267, 285, 335

- adding a segment • 267
- BYTES= • 73, 81
- DATA SET • 73
- DATA SET statement • 73
- DATA SET= • 73
- DD1= • 73
- description of • 65, 70
- DEVICE= • 73
- FIELD statement • 73, 77, 81, 84, 90, 285, 335
- importing • 65
- last update of • 70
- LCHILD • 73
- maintaining a • 73
- name of • 65, 77, 84, 90, 95
- NAME= • 73, 77, 81, 84, 90, 285, 335
- PARENT= • 73, 81
- SEGM • 73
- SEGM statement • 73, 81
- START= • 73, 81
- DBD field • 84, 90, 95
 - Create/Update SSA/Command For DL/I DB/Segment • 90
 - Update DLIDSCs For Segment Member • 84
 - Verify DLIDSC Count • 95
- DBINFO • 47
 - DBINFO • 47
- DBLSPC field • 534
 - Utilities menu • 534
- DBNAME field • 102, 170, 189, 199, 448, 526
 - Update Batch Output Fields • 448
 - Update Output/Input/Outin Field • 189
 - Update Panel Fields (Online) • 170
 - Update Select Fields • 199
 - Update Sensitive Segment • 102
 - View Presentation Store • 526
- DBNAME OR TEXT field • 442
 - Update Panel Fields (Batch) • 442
- DCLCOPY field • 124
 - Create/Update SQL Tables/TLNROWS • 124
- DCLLBL field • 124
 - Create/Update SQL Tables/TLNROWS • 124
- DCLRDEF field • 124
 - Create/Update SQL Tables/TLNROWS • 124
- DEBUG field • 45
 - Update Environment Definition Defaults • 45
- DEC • 132
 - Create/Update IDMS SQL Tables/TLNROWS • 132
- DEC field • 124, 332
 - Create/Update SQL Tables/TLNROWS • 124
 - Select Columns • 332
- DEFAULT HEADER field • 55
 - Update Session Controls • 55
- defaults • 29, 30, 53, 55
 - PF keys • 53
 - session controls • 55
 - Update Program Definition Defaults • 30
 - User Profile Maintenance • 29
- DEFIN field • 156
 - Panel Definition menu • 156
- DELETE field • 298
 - Update SQL Detail Data Access • 298
- Delimiters • 17
 - syntax diagrams, use in • 17
- DESC field • 65, 156, 246, 250, 401, 416, 429, 458, 462, 534
 - Create/Update Screen Definition • 250
 - Data Administration menu • 65
 - Online Program Definition menu • 246
 - Panel Definition menu • 156
 - Utilities menu • 534
- DESCR • 132, 141
 - Create/Update IDMS SQL Tables/TLNROWS • 132
 - Update SQL ALIAS • 141
- DESCR field • 124
 - Create/Update SQL Tables/TLNROWS • 124
- DESCRIPTION field • 70, 122, 162, 209, 368, 481, 483, 520, 523
 - List Consistency Edits • 209
 - List Data Administration Information • 70
 - List Panel Definitions • 162, 520
 - List Presentation Stores • 523
 - List SQL Tables/Joins • 122
- DETAIL • 442, 454
 - GPTYPE field • 442, 454
- DETERM • 47
 - DETERM • 47
- DG SEPARATORS field • 55
 - Update Session Controls • 55
- DGADD • 348
 - queues • 348
- DGADDING • 117
 - journals • 117
- DICTNAM • 132, 141
 - Create/Update IDMS SQL Tables/TLNROWS • 132

- Update SQL ALIAS • 141
- DITTO CHARACTER field • 55
 - Update Session Controls • 55
- DL/I • 37, 45
 - automatic weighting of calls • 45
 - PCB masks • 37
 - PSBs, scheduling and terminating • 37
- DLI QUALIFY field • 215
 - Update Segment Edit (SEGEDIT) • 215
- DLIDSC • 84, 285, 335
 - selecting • 84, 285, 335
- DLIDSC field • 77, 84, 90, 95
 - Create/Update SSA/Command For DL/I DB/Segment • 90
 - Update DBD Segment • 77
 - Update DLIDSCs For Segment Member • 84
 - Verify DLIDSC Count • 95
- DLIWGHT field • 45, 475
 - Update Environment Definition Defaults • 45
- driver definition • 246
 - creating/updating/purging/showing a • 246
 - description of • 246
- DSCREF field • 335
 - Update DL/I Detail Data Access • 335

E

- EACOLOR field • 180, 185, 189, 199
 - Update Attribute Parameter • 180
 - Update Literal Fields • 185
 - Update Output/Input/Outin Field • 189
 - Update Select Fields • 199
- EAERR field • 30, 394
 - Update Program Definition Defaults • 30
- EAHIGH field • 180, 185, 189, 199
 - Update Attribute Parameter • 180
 - Update Literal Fields • 185
 - Update Output/Input/Outin Field • 189
 - Update Select Fields • 199
- EAIN field • 30, 394
 - Update Program Definition Defaults • 30
- EALIT field • 30, 394
 - Update Program Definition Defaults • 30
- EAOUT field • 30, 394
 - Update Program Definition Defaults • 30
- EATTR field • 30, 382, 394
 - Update Program Definition Defaults • 30
- EAVAILD field • 180, 189, 199
 - Update Attribute Parameter • 180
 - Update Output/Input/Outin Field • 189

- Update Select Fields • 199
- EDIT CONDITION field • 212
 - Update Cross-Field Edit (XFEDIT) • 212
- Edit Flag fields • 298, 429
 - Update SQL Detail Data Access • 298
- EDIT NAME field • 212, 215
 - Update Cross-Field Edit (XFEDIT) • 212
 - Update Segment Edit (SEGEDIT) • 215
- ENV FORMAT field • 534
 - Utilities menu • 534
- ENV PSB field • 534
 - Utilities menu • 534
- ENVIRON field • 246, 458
 - 4.0B Online Program Definition menu • 246
- ENVIRONMENT field • 55, 385
 - Update Session Controls • 55
- EOFKEY field • 30, 394
 - Update Program Definition Defaults • 30
- erase fields • 30
 - with ERASE EOF key • 30
- ERRCHAR FIELDS field • 212, 215
 - Update Cross-Field Edit (XFEDIT) • 212
 - Update Segment Edit (SEGEDIT) • 215
- ERROR CONDITION field • 215
 - Update Segment Edit (SEGEDIT) • 215
- ERROR MESSAGE field • 212, 215
 - Update Cross-Field Edit (XFEDIT) • 212
 - Update Segment Edit (SEGEDIT) • 215
- EXP field • 96
 - Create/Update PSB Or File Group • 96
- EXPRESS field • 277
 - Update DBMS Characteristics • 277
- extended field list • 207
 - input edits • 207
 - output edits • 207
- EXTSCUR • 47
 - EXTSCUR • 47

F

- FDREC field • 343
 - Update VSAM or Sequential Detail Data Access • 343
- FENCED • 47
 - FENCED • 47
- FGNAME field • 104
 - Update Sensitive TLNROWS • 104
- field • 81, 448
 - CONVERT • 448
 - FORMAT • 448

- Type • 81
- FIELD field • 156
 - Panel Definition menu • 156
- FIELD NAME field • 185, 189, 199, 207, 446, 448
 - Update Batch Literal Fields • 446
 - Update Batch Output Fields • 448
 - Update Literal Fields • 185
 - Update Output/Input/Outin Field • 189
 - Update Parameter List Extension • 207
 - Update Select Fields • 199
- FIELD NAME OR POSITION field • 207
 - Update Parameter List Extension • 207
- FIELD SELECTION field • 55
 - Update Session Controls • 55
- field types • 30
 - APPLID field • 30
 - OUTIFIL field • 30
- FIELD UPDATE field • 120
 - Catalog/Import DB2 Tables • 120
- field update rules • 542, 544, 545, 547, 548, 549
 - fields with panel data • 545
 - general rules • 542
 - literal fields • 544
 - long literal fields • 549
 - panel editors • 542
 - variable fields • 544
 - wrapped literal fields • 548
 - wrapped variable fields • 547
- fields • 25, 29, 30, 36, 37, 40, 44, 45, 53, 55, 65, 70, 73, 77, 81, 84, 90, 95, 96, 102, 104, 106, 113, 117, 120, 122, 124, 132, 141, 143, 145, 147, 150, 153, 156, 162, 165, 168, 170, 176, 178, 180, 185, 189, 199, 207, 209, 212, 215, 228, 233, 246, 250, 267, 277, 281, 285, 291, 298, 327, 332, 335, 341, 343, 348, 352, 356, 361, 365, 368, 371, 373, 382, 385, 387, 394, 401, 410, 416, 423, 426, 429, 437, 442, 446, 448, 454, 458, 462, 475, 479, 481, 483, 486, 493, 514, 520, 523, 526, 529, 534, 540, 541
 - &caret.N (NOTNULL) • 124, 132
 - &caret.NULL • 332
 - (Command field) • 529
 - (Function) • 162
 - (KEY extension) • 285
 - (Line command) • 84, 145, 147, 267, 335, 371, 481, 483, 520, 523, 526
 - (Line commands) • 168
 - (Message area) • 120
 - (Panel image display window) • 176, 178, 180
 - (Select field) • 122
 - (Sequence number) • 77, 81
 - (TO) ID • 534
 - (Update field) • 77
- ABC • 96
- ABCALL • 277
- AC • 124, 132, 147, 150
- AC (ACCESS) • 332
- ACCESS • 73, 106, 291
- ACCESS COLUMNS • 145
- ACCMODE • 106, 291
- ALARM • 30, 394
- ALIAS • 124, 132, 141, 147, 150, 332
- APPLID • 30, 394, 401, 416, 429, 462
- ATTRINT • 180, 185, 189, 199
- ATTRPRO • 180, 189, 199
- AUX/MAIN • 113, 348, 356
- BASE • 65
- BASE DEFN • 246, 458
- BLANK.WHEN.SAME • 446, 448
- BLKSIZ • 291
- BLKSIZE • 106
- BMS • 37, 387
- BOOLOP • 90
- CALC • 448
- CALL FUNC • 215
- CAPS • 30, 394
- CASE • 156
- C-CNT • 104, 281
- CINCRE • 228, 233
- CJNAME • 117, 352
- CMDCODE • 90, 215
- CMND • 77, 84, 285, 335
- CNTGRP • 448
- COBDIV • 106, 291
- COBVSky • 106, 291
- COL • 170, 176, 178, 180, 426
- COLS • 120
- COLSGLP • 228, 233
- COLUMN • 228, 233
- COLUMN NAME • 124, 132, 141, 145, 147, 150, 332
- COLUMNS DEFINED • 104, 281
- COMMAND • 132, 141, 150
- COMMIT • 106

COMPAT • 96
COMPRESS STATEMENTS • 55
COMPRESSED • 534
CONCATK • 90, 335
CONSIS • 156, 250
CONVERS • 40, 373, 410
COPY • 77, 106, 113, 117, 124, 132, 141, 285, 291, 348, 352, 486
COPY EDIT BASE • 212, 215
COPYLBL • 77, 106, 113, 117, 124, 132, 141, 285, 291, 348, 352, 486
COPYLV1 • 77, 106, 113, 117, 124, 132, 141, 285, 291, 348, 352, 486
CORRELATION NAME • 143, 145, 147, 150
CPYCALL • 298
CPYINIT • 298
CPYKEY • 298
CPYTERM • 298
CQNAME • 113, 348
CURRENT • 90, 335
CURRENT ROW NAME • 281
CURSCUS • 250
CURSOR • 250
CURSOR AT FIELD • 212, 215
CUSTCODE • 246, 458
DATA SET • 106, 291, 493
DBD • 84, 90, 95, 285
DBLSPC • 534
DBMS • 45, 387, 475
DBNAME • 102, 189, 199, 448, 526
DBNAME OR TEXT • 170, 442
DCLCOPY • 124
DCLLBL • 124
DCLRDEF • 124
DEC • 124, 132, 332
DEFAULT HEADER • 55
DEFIN • 156
DELETE • 298
DESC • 65, 156, 246, 250, 401, 416, 429, 458, 462, 534
DESCR • 124, 132, 141
DESCRIPTION • 70, 122, 162, 209, 368, 481, 483, 520, 523
DG SEPARATORS • 55
DICTNAM • 132, 141, 332
DITTO CHARACTER • 55
DLI QUALIFY • 215
DLIDSC • 77, 84, 90, 95
DLIWGHT • 45, 475
DSCREF • 285, 335
EACOLOR • 180, 185, 189, 199
EAERR (color attribute) • 30, 394
EAERR (highlight attribute) • 30, 394
EAHIGH • 180, 185, 189, 199
EAIN (color attribute) • 30, 394
EAIN (highlight attribute) • 30, 394
EALIT (color attribute) • 30, 394
EALIT (highlight attribute) • 30
EAOUT (color attribute) • 30, 394
EAOUT (highlight attribute) • 30, 394
EATTR • 30, 382, 394
EAVAILD • 180, 189, 199
EDIT CONDITION • 212
Edit Flag fields • 298
EDIT NAME • 212, 215
ENV FORMAT • 534
ENV PSB • 534
ENVIRON • 246, 458
ENVIRONMENT • 55, 385, 534
EOFKEY • 30, 394
ERRCHAR FIELDS • 212, 215
ERROR CONDITION FOUND/NOTFOUND • 215
ERROR MESSAGE • 212, 215
EXP • 96
EXPRESS • 277
FDREC • 343
FGNAME • 104
FIELD • 156
FIELD NAME • 185, 189, 199, 446, 448
FIELD NAME OR POSITION • 207
FIELD SELECTION • 55
FIRST • 228, 233
FLDEDIT • 250
FLDTYPE • 170, 189, 199, 448
FMTCNTL • 189, 199
FMTEXT • 189, 199
FORMAT • 55
FROM • 156
FROM QUAL.TABLE • 147
FTCHOPT • 298
FUNC • 335, 341, 343, 361, 365
FUNCTION • 25, 29, 65, 70, 156, 246, 458, 514, 534
GEN KEY LTH • 215
GENKEYL • 106, 291, 343
GENPCBS • 37, 40, 44, 45, 373, 387, 410, 423, 475

GET • 40
GETDIAG • 298
GPTYPE OR FLDTYPE • 442
GROUP • 448, 458
GROUP NAME • 454
GROUPBY • 298
HAVING • 298
HEADER • 156, 246, 458, 514, 534
HELP • 30, 394
HELPMMSG • 176, 189, 199
HIGHLIGHT FIELDS • 212, 215
HOLD • 30, 394, 401
HOLD CURSOR • 298
I/O AREA • 215
ICTLNM • 228, 233
ICUST1 • 228, 233
ICUST2 • 228, 233
ID • 156, 246, 458, 514, 534
IEXTEND • 207
IGNORE • 267, 298, 335, 341, 343, 356, 361
IMAGE • 156
IMAGE CHARACTERS • 55
IMSKEY • 77, 81, 84, 90, 285, 335
INCRE • 228, 233
INDBIO • 199, 426
INDEXOF • 106, 291
INDICES • 102
INEDIT • 199, 426
ININIT1 • 250
ININIT2 • 250
INIT • 189, 199, 401, 448
INTERM • 250
IOAREA • 298, 335, 341, 343, 356, 361
IOASTG • 37, 387
ISEGIDX • 228, 233
ITEM • 65, 156, 246, 458, 514
JFILEID • 117, 352, 361
JOIN COLUMNS • 147
JOINOPT • 298
JTYPEID • 117, 352, 361
KEY • 77, 84, 90, 106, 215, 285, 291, 298, 335, 343
KEY OR WHERE • 281
KEY/WHERE • 267
KEYCOLS • 298
KEYFEED • 90
KEYLEN • 96, 106, 285, 291
KEYLTH • 277
KEYPIC • 81, 285
KY • 124, 132, 150
KY (KEY) • 332
LABEL • 77, 102, 106, 113, 117, 267, 285, 291, 348, 352, 382
LANG • 30, 96, 250, 401, 416, 429, 462
LAST • 228, 233
LENGTH • 73, 81
LINE • 228, 233, 426
LINE COL • 170, 176, 178, 180, 442
LINE COL LTH • 185, 189, 199, 446, 448
Line command • 132
LINECNT • 228, 233
LINENUM • 534
LINEOPT • 37, 40, 373, 387
LINKOPT • 40, 373
LN • 170, 176, 178, 180, 426
LN COL LTH • 442
LNKCOPY • 37, 40, 44, 45, 373, 387, 410, 423, 475
LOCKED • 90, 335
LRECL • 106, 113, 117, 291, 348, 352, 486
LTERM • 96, 277
LTH • 124, 132, 170, 176, 178, 180, 332, 382, 426, 479, 483, 486, 526
LTHLBL • 348, 352, 356, 361
LTHOPT • 348, 352, 356, 361
MAPOUT • 178, 189, 448
MAX LTH • 73, 81
MORE • 84, 170, 442, 479, 486
MSGCALL • 277
NAME • 65, 70, 73, 81, 96, 102, 156, 162, 170, 176, 178, 180, 368, 426, 442, 523
NEXTPGM • 199, 250, 426
OCUST1 • 228, 233
OCUST2 • 228, 233
OCUST3 • 228, 233
OEXTEND • 207
OF • 189, 199, 448
OFFSET • 90
OINIT1 • 250, 416
OINIT2 • 250, 416
OP • 77, 84, 90, 285, 335
OPCODE • 106, 215, 291, 298, 343
OPEN • 106, 291
OPTION • 90
OPTIONS • 170, 250, 401, 416, 429, 462
OPTLIST • 106, 291, 343
ORDERBY • 298
OSEGIDX • 228, 233

OUTATTR • 180, 185, 189
OUTIFIL • 30, 394
OUTPUT FIELD • 281
OUTPUT LINE • 143
OUTTERM • 250, 416
PAGE • 233
PAGEKEY • 233
PAGESAV • 233
Panel image display window • 170
PANEL SIZE • 55
PARENT • 81
PARENT/DEVICE • 73
PARENTG • 90, 335
PARM LIST EXTENSION • 189, 199, 448
PATH • 90, 335
PCB • 277
PCB PARMS • 77, 285
PCBNAME • 96, 102, 215, 277
PDSCOPY DSNNAME • 55
PF/PA KEY DEFINITIONS • 53
PFKEYS • 250
PFXLBL • 352, 361
PFXLTH • 352, 361
PGMCUST • 30, 250, 401, 416, 429, 462
PI/PD NAME • 520
PIC • 189, 448
PKYLTH • 233
PKYUNIQ • 233
PR STORE/RENAME • 520
PRESENTATION • 514
PREVIEW • 298, 335, 341, 343, 356, 361
PRINT • 277
PROCOPT • 96, 102, 277
PROCSEQ • 90, 96, 277
PRT (PRINT) • 96
PS NAME • 514
PSB • 55
PSBNAME • 37, 387, 475
PSBSCHD • 37, 387
PUT • 40
QUAL • 120, 143
QUELBL • 348
RDBMS • 141
READSET • 343
RECLTH • 106, 291, 343
REMARKS • 250, 401, 416, 429, 462
RENAME • 70, 162, 368, 523
REPEAT • 228, 233
REQ • 170, 189, 442

REQUEST • 267
RESLTPR • 298
REUSE • 106
RMNAME • 73
ROW NAME • 132, 332
S • 332
SAVEKEY • 228, 233
SCHEMA • 132, 141, 332
SCHFLDC • 233
SCHFLDI • 233
SCHFLDL1 • 233
SCONSIS • 199, 426
SCOPE • 448
SCREEN XFEDIT/SEGEDIT • 250
SEARCH FIELDS • 77
SECTION • 250, 401, 416, 429, 462
SEGLOOP • 156, 228, 233
SEGLOOP DELIMITERS • 170
SEGM • 77, 285
SEGMENT • 77, 84, 90, 95, 285, 335, 341
SEGMENT KEY • 73
SEGMENT NAME • 215
SELECT • 281, 462
SELECT FIELDS • 250
SELKEY • 199
SENCOLS • 298
SEQ • 73, 96, 102, 104, 209, 479, 486
SET • 356
SHIFT • 534
SIZE • 156, 250, 416, 426, 429, 442, 462, 479, 483, 486
SKIPAFT • 454
SKIPBEF • 454
SPACMPT • 40, 373
SPASIZE • 37, 40, 373, 387, 410
SPASTG • 37, 387
SPEC • 189, 199, 448
SQL QUAL.NAME • 122
SRCHFLD NAME • 81
SSA LIST HOST VARIABLE • 341
SSALIST • 215, 335, 341
START • 73, 81, 486
STBRKEY • 233
SUB • 526
SYNONYM • 124, 132, 277
SYSID • 113, 348, 356
TABLE • 150, 277
TABLE NAME • 143, 281, 332
TABLE-NAME • 120

TBLNAME • 104
 TEXT • 185, 446
 TLN • 120
 TLNNAME • 120, 122, 124, 132, 141, 143, 145, 147, 150, 277
 TLNROW • 104, 141, 281
 TMPCOMT • 281
 TMPNAME • 281
 TMPQUAL • 281
 TO • 156
 TO HEADER • 534
 TOTREF • 448
 TOTSIZ • 448
 TPBSTG • 37, 387
 TRACE • 37, 40, 45, 373, 387, 410, 423, 475
 TRANCDE • 40, 373, 387, 410
 TRANFLD • 40, 373
 TRANMFS • 40, 373
 TYPE • 73, 81, 96, 106, 113, 124, 132, 209, 246, 277, 291, 332, 348, 356, 382, 454, 534
 U • 90, 170, 442
 UPDATE • 70, 120, 162, 298, 520, 523
 UPDATE DBD • 77
 UPDATE ENVIRON DEFN DEFAULTS • 30
 UPDCOLS • 327
 UPPER CASE LITERALS • 55
 USAGE • 185, 189, 199, 281, 285, 291, 348, 352, 446, 448
 USE • 170, 176, 178, 180, 442
 USECNT • 84
 USECOL1 • 534
 USED BY • 95
 USER • 70, 162, 368, 520, 523
 USER MODE • 55
 USERID • 120
 USGCOP1 • 37
 USGCOP2 • 37
 USGCOPY • 40, 44, 45, 373, 410, 423, 475
 VALUE • 526
 VARLTH • 90
 VSAM SEGMENT LTH • 215
 WHEN • 215
 WHERE • 298
 WKAREA • 250, 401, 416, 429, 462
 WKSPAIN • 40, 373, 410
 WKSPASZ • 40, 373, 410
 XFERWKA • 30, 250, 401, 416, 429
 file • 65, 70, 199
 description of • 65, 70
 field name for mapping data • 199
 last update of • 70
 name of • 65
 File Group • 65, 70, 96
 defining/maintaining a • 96
 description of • 65, 70
 last update of • 70
 name of • 65
 file SEGLOOP • 233
 Create/Update File SEGLOOP • 233
 FIRST field • 228, 233
 Create/Update File SEGLOOP • 233
 Create/Update Table SEGLOOP • 228
 FLDEDIT field • 250
 Create/Update Screen Definition • 250
 FLDTYPE field • 170, 189, 199, 442, 448, 513
 Update Batch Output Fields • 448
 Update Output/Input/Outin Field • 189
 Update Panel Fields (Batch) • 442
 Update Panel Fields (Online) • 170
 Update Select Fields • 199
 use with prototyping • 513
 FMTCNTL field • 189, 199
 Update Output/Input/Outin Field • 189
 Update Select Fields • 199
 FMTEXTIT field • 189, 199
 Update Output/Input/Outin Field • 189
 Update Select Fields • 199
 FORMAT • 448
 FIELD • 448
 FORMAT field • 55
 Update Session Controls • 55
 FROM field • 156
 Panel Definition menu • 156
 FROM QUAL.TABLE field • 147
 Update SQL Join? Access Columns • 147
 use with ALIAS field • 147
 full screen edit mode • 542
 invoking • 542
 FUNC field • 120, 335, 343, 361, 365
 DB2/Catalog Import • 120
 Update DL/I Detail Data Access • 335
 Update VSAM or Sequential Detail Data Access • 343
 FUNCTION field • 25, 29, 70, 156, 162, 246, 341, 458, 514, 534
 List Data Administration Information • 70
 List Panel Definitions • 162

- Online Program Definition menu • 246
- Panel Definition menu • 156
- Prototyping Facility menu • 514
- TDF Main menu • 25
- Update Database User I/O SSALIST • 341
- User Profile Maintenance • 29
- Utilities menu • 534

G

- GEN KEY LTH field • 215
 - Update Segment Edit (SEGEDIT) • 215
- GENKEYL field • 106, 291, 343
 - Show/Update Data Set Default Data • 106
 - Update Data Set Record • 291
 - Update VSAM or Sequential Detail Data Access • 343
- GENPCBS • 37
 - Update Environment Definition Defaults • 37
- GENPCBS field • 40, 44, 45, 373, 387, 410, 423, 475
 - Update Environment Definition Defaults • 40, 44, 45
- GET DIAGNOSTICS • 309, 312, 316, 320, 323
 - Combined information screen • 320
 - Condition information screen • 316
 - List Screen • 309
 - Statement information screen • 312
 - statement values • 309, 312, 316, 320
- GETDIAG field • 298
 - Update SQL Detail Data Access • 298
- GPTYPE OR FLDTYPE field • 442
 - Update Panel Fields (Batch) • 442
- GROUP field • 448, 458
 - Update Batch Output Fields • 448
- GROUP NAME field • 454
 - Update Panel Group • 454
- GROUPBY field • 298
 - Update SQL Detail Data Access • 298

H

- H-100 program section • 199, 426
 - create/update segments for Auto Exec • 199
- handle conditions • 356, 361
 - CICS • 356, 361
- HAVING field • 298
 - Update SQL Detail Data Access • 298
- HEADER field • 156, 246, 458, 514, 534
 - Online Program Definition menu • 246
 - Panel Definition menu • 156

- Prototyping Facility menu • 514
- Utilities menu • 534
- HELP field • 30, 394
 - Update Program Definition Defaults • 30
- help, online • 13
 - for TDF screens, fields • 13
- HELPMMSG field • 176, 189, 199
 - Update Help Message Parameters • 176
 - Update Output/Input/Outin Field • 189
 - Update Select Fields • 199
- highlight attribute • 30, 180, 394
 - EAHIGH field • 180
 - Update Program Definition Defaults • 30
- HIGHLIGHT field • 212
 - Update Cross-Field Edit (XFEDIT) • 212
- HIGHLIGHT FIELDS field • 215
 - Update Segment Edit (SEGEDIT) • 215
- HOLD command • 513
 - prototyping facility restriction • 513
- HOLD CURSOR field • 298
 - Update SQL Detail Data Access • 298
- HOLD field • 30, 394, 401
 - Update Program Definition Defaults • 30

I

- I/O area • 77, 106, 124, 285, 291, 298, 332
 - column • 124, 332
 - default • 77, 106, 124, 285, 291
 - defining the • 298
 - record • 106, 291
 - segment • 77, 285
 - SQL • 124
- I/O AREA field • 215
 - Update Segment Edit (SEGEDIT) • 215
- I/O area, record • 106, 291
 - overriding of • 106, 291
- I/O area, segment • 77, 285
 - overriding of • 77, 285
- I/O request • 90
 - qualifying a • 90
- ICTLNM field • 228, 233
 - Create/Update File SEGLOOP • 233
 - Create/Update Table SEGLOOP • 228
- ICUST1 field • 228, 233
 - Create/Update File SEGLOOP • 233
 - Create/Update Table SEGLOOP • 228
- ICUST2 field • 228, 233
 - Create/Update File SEGLOOP • 233
 - Create/Update Table SEGLOOP • 228

ID field • 156, 246, 458, 514, 534
 Online Program Definition menu • 246
 Panel Definition menu • 156
 Prototyping Facility menu • 514
 Utilities menu • 534
IDENTIF field • 250, 401, 416, 429, 462
 Create/Update Screen Definition • 250
IEXTEND field • 207
 Update Parameter List Extension • 207
IGNCLOS field • 291
 Update Data Set Record • 291
IGNEMPT field • 291
 Update Data Set Record • 291
IGNOPEN field • 291
 Update Data Set Record • 291
IGNORE field • 267, 298, 335, 341, 343, 356, 361
 Create/Update Data Group • 267
 Update CICS Queue Data Access • 356
 Update Database User I/O SSALIST • 341
 Update DL/I Detail Data Access • 335
 Update SQL Detail Data Access • 298
 Update VSAM or Sequential Detail Data Access • 343
IMAGE CHARACTERS field • 55
 Update Session Controls • 55
IMAGE field • 156
 Panel Definition menu • 156
import • 120
 date of • 120
 DB2 tables • 120
 DB2 views • 120
 error condition • 120
IMS • 40, 55
 conversational system, generating a • 40
 environment • 55
 environment fields • 40
 generating programs • 40
 linking in a program • 40
 SPA • 40
 transaction code • 40
IMSKEY field • 77, 81, 84, 90, 285, 335
 Create/Update SSA/Command For DL/I DB/Segment • 90
 List Search Fields • 81
 Update Database Segment • 285
 Update DBD Segment • 77
 Update DL/I Detail Data Access • 335
 Update DLIDSCs For Segment Member • 84
INCRE field • 228
 Create/Update Table SEGLOOP • 228
INDBIO field • 199, 426
 Update Select Fields • 199
INDEXOF field • 106, 291
 Show/Update Data Set Default Data • 106
 Update Data Set Record • 291
INEDIT field • 199, 426
 Update Select Fields • 199
ININIT1 field • 250
 Create/Update Screen Definition • 250
ININIT2 field • 250
 Create/Update Screen Definition • 250
INIT field • 189, 199, 401, 448
 Update Batch Output Fields • 448
 Update Output/Input/Outin Field • 189
 Update Select Fields • 199
input field • 199, 228, 233
 Create/Update File SEGLOOP • 233
 Create/Update Table SEGLOOP • 228
 mapping and editing • 199
input values • 189, 448
 acceptable • 189, 448
 how the program stores them • 189, 448
input, field type • 189, 212, 215, 448
 acceptable input values • 189, 448
 CURSOR AT FIELD field • 212, 215
 required • 212
 screen field entries • 215
INTENSITY • 514, 520
 prototyping facility • 514
INTERM field • 250
 Create/Update Screen Definition • 250
IOAREA field • 298, 335, 341, 356, 361
 Update CICS Queue Data Access • 356
 Update Database User I/O SSALIST • 341
 Update DL/I Detail Data Access • 335
 Update SQL Detail Data Access • 298
IOASTG field • 37, 387
 Update Environment Definition Defaults • 37
ISEGIDX field • 228, 233
 Create/Update File SEGLOOP • 233
 Create/Update Table SEGLOOP • 228
ITEM field • 65, 156, 246, 458, 514
 Data Administration menu • 65
 Online Program Definition menu • 246
 Panel Definition menu • 156
 Prototyping Facility menu • 514

J

- JFILEID field • 117, 352, 361
 - Update CICS Journal Default Data • 117
 - Update CICS Journal Record • 352
- join • 65, 120, 122, 143, 145, 147
 - correlation name • 143, 145, 147
 - description of • 65
 - identifier • 145
 - key name • 145
 - name of • 65, 122
 - naming a • 120, 143, 147
 - qualifier • 143
 - WHERE clause • 145, 147
- JOIN COLUMNS field • 147
 - Update SQL Join? Access Columns • 147
- join tables • 147
 - defining access for • 147
- JOINOPT field • 298
 - Update SQL Detail Data Access • 298
- journals • 65, 117, 120, 267, 352, 361
 - accesses, group level variable name • 117
 - COBOL COPY, overriding • 352
 - defining • 65
 - DGADding • 117
 - IDs, valid • 117, 352
 - inheritance of name • 352
 - length of record • 117, 352, 361
 - PL/I INCLUDE, overriding • 352
 - prefix for • 352, 361
 - REC line, use of • 267
 - record, defining • 117
 - types of • 352
- JTYPEID field • 117, 352, 361
 - Update CICS Journal Default Data • 117
 - Update CICS Journal Record • 352

K

- key • 106, 215, 291, 343
 - generic, length of • 106, 291, 343
 - identifying the record/segment to be read • 215
 - relative • 106, 291
 - variable • 106, 291
- KEY field • 77, 84, 90, 106, 215, 285, 291, 298, 335
 - Create/Update SSA/Command For DL/I DB/Segment • 90
 - Show/Update Data Set Default Data • 106
 - Update Data Set Record • 291

- Update Database Segment • 285
- Update DBD Segment • 77
- Update DL/I Detail Data Access • 335
- Update DLIDSCs For Segment Member • 84
- Update Segment Edit (SEGEDIT) • 215
- Update SQL Detail Data Access • 298
- KEY/WHERE field • 267, 281
 - Create/Update Data Group • 267
 - Select New Row Name • 281
- KEYCOLS field • 298
 - Update SQL Detail Data Access • 298
- KEYFEED field • 90
 - Create/Update SSA/Command For DL/I DB/Segment • 90
- KEYLEN field • 96, 106, 285, 291
 - Create/Update PSB Or File Group • 96
 - Show/Update Data Set Default Data • 106
 - Update Data Set Record • 291
 - Update Database Segment • 285
- KEYLTH field • 277
 - Update DBMS Characteristics • 277
- KEYPIC field • 81, 285
 - List Search Fields • 81
 - Update Database Segment • 285
- Keys • 77
 - segment, retrieving • 77
- KY • 132, 150
 - Add/Update SQL Join Alias • 150
 - Create/Update IDMS SQL Tables/TLNROWS • 132
- KY field • 147, 332
 - Select Columns • 332
 - Update SQL Join? Access Columns • 147

L

- LABEL field • 77, 102, 106, 113, 117, 267, 285, 291, 348, 352, 382
 - Create/Update Data Group • 267
 - Create/Update PSB Or File Group • 102
 - Show/Update Data Set Default Data • 106
 - Update CICS Journal Default Data • 117
 - Update CICS Journal Record • 352
 - Update CICS Queue Default Data • 113
 - Update CICS Queue Record • 348
 - Update Data Set Record • 291
 - Update Database Segment • 285
 - Update DBD Segment • 77
- LANG field • 30, 96, 250, 401, 416, 429, 462
 - Create/Update PSB Or File Group • 96

- Create/Update Screen Definition • 250
- Update Program Definition Defaults • 30
- language • 30
 - specifying for the program • 30
- LAST field • 228, 233
 - Create/Update File SEGLOOP • 233
 - Create/Update Table SEGLOOP • 228
- LENGTH field • 73, 81
 - Create/Update DBD • 73
 - List Search Fields • 81
- LINE COL field • 170, 176, 178, 180, 442
 - Update Attribute Parameter • 180
 - Update Help Message Parameters • 176
 - Update Mapout Parameter • 178
 - Update Panel Fields (Batch) • 442
 - Update Panel Fields (Online) • 170
- Line command • 132
 - Create/Update IDMS SQL Tables/TLNROWS • 132
- line edit mode • 168, 542
 - command line • 168
 - invoking • 542
 - panel image • 168
- LINE field • 185, 189, 199, 228, 233, 426, 442, 446, 448
 - Create/Update File SEGLOOP • 233
 - Create/Update Table SEGLOOP • 228
 - Update Batch Literal Fields • 446
 - Update Batch Output Fields • 448
 - Update Literal Fields • 185
 - Update Output/Input/Outin Field • 189
 - Update Panel Fields (Batch) • 442
 - Update Select Fields • 199
- LINECNT field • 228
 - Create/Update Table SEGLOOP • 228
- LINENUM field • 534
 - Utilities menu • 534
- LINEOPT • 40
 - Update Environment Definition Defaults • 40
- LINEOPT field • 37, 373, 387
 - Update Environment Definition Defaults • 37
- LINKOPT • 40
 - Update Environment Definition Defaults • 40
- List Consistency Edits • 222
 - consistency edits example • 222
- literal break character • 544
 - literal field rules • 544
- literal field • 156, 442
 - translate to upper case • 156
 - type of editing • 442
- literal, field type • 55
 - converting to uppercase • 55
- LN field • 170, 176, 178, 180, 426, 442
 - Update Attribute Parameter • 180
 - Update Help Message Parameters • 176
 - Update Mapout Parameter • 178
 - Update Panel Fields (Batch) • 442
 - Update Panel Fields (Online) • 170
- LNKCOPY field • 37, 40, 44, 45, 373, 387, 410, 423, 475
 - including 01-level declarations • 37
 - Update Environment Definition Defaults • 37, 40, 44, 45
- LOCKED field • 90, 335
 - Create/Update SSA/Command For DL/I DB/Segment • 90
 - Update DL/I Detail Data Access • 335
- logical record length • 291
 - maximum length • 291
- lower case to upper case • 514
 - prototyping facility • 514
- LRECL field • 106, 113, 117, 291, 348, 352, 486
 - Show/Update Data Set Default Data • 106
 - Update CICS Journal Default Data • 117
 - Update CICS Journal Record • 352
 - Update CICS Queue Default Data • 113
 - Update CICS Queue Record • 348
 - Update Data Set Record • 291
- LTERM field • 96, 277
 - Create/Update PSB Or File Group • 96
 - Update DBMS Characteristics • 277
- LTH • 132
 - Create/Update IDMS SQL Tables/TLNROWS • 132
- LTH field • 170, 176, 178, 180, 185, 189, 199, 332, 382, 426, 442, 446, 448, 479, 483, 486, 526
 - Select Columns • 332
 - Update Attribute Parameter • 180
 - Update Batch Literal Fields • 446
 - Update Batch Output Fields • 448
 - Update Help Message Parameters • 176
 - Update Literal Fields • 185
 - Update Mapout Parameter • 178
 - Update Output/Input/Outin Field • 189
 - Update Panel Fields (Batch) • 442
 - Update Panel Fields (Online) • 170
 - Update Select Fields • 199

- View Presentation Store • 526
- LTHLBL field • 348, 352, 356, 361
 - Update CICS Journal Record • 352
 - Update CICS Queue Data Access • 356
 - Update CICS Queue Record • 348
- LTHOPT field • 348, 352, 356, 361
 - Update CICS Journal Record • 352
 - Update CICS Queue Data Access • 356
 - Update CICS Queue Record • 348

M

- main storage • 113, 348, 356, 486
 - queues • 113, 348, 356
- MAPOUT field • 178, 189, 448
 - Update Batch Output Fields • 448
 - Update Mapout Parameter • 178
 - Update Output/Input/Outin Field • 189
- mapping, output • 178
 - MAPOUT field • 178
- match sort • 267, 462, 479
 - AUTOEXEC requests • 267
 - USEREXEC requests • 267
- MAX LTH field • 73, 81
 - Create/Update DBD • 73
 - List Search Fields • 81
- member • 534
 - copying • 534
 - list • 534
 - renaming • 534
- merge presentation-store • 529
 - into active presentation store • 529
- MFS • 170, 185, 189, 199, 373, 382, 423
 - control blocks, generation of • 170, 185, 189, 199
 - field exit routine • 189, 199
 - special field • 189, 199
- MORE field • 84, 170, 442, 479, 486
 - Update DLIDSCs For Segment Member • 84
 - Update Panel Fields (Batch) • 442
 - Update Panel Fields (Online) • 170
- MSGCALL field • 277
 - Update DBMS Characteristics • 277

N

- NAME field • 65, 73, 81, 96, 102, 156, 162, 170, 176, 178, 180, 368, 426, 442, 520, 523
 - Create/Update DBD • 73
 - Create/Update PSB Or File Group • 96
 - Data Administration menu • 65

- List Panel Definitions • 162, 520
- List Presentation Stores • 523
- List Search Fields • 81
- Panel Definition menu • 156
- Update Attribute Parameter • 180
- Update Help Message Parameters • 176
- Update Mapout Parameter • 178
- Update Panel Fields (Batch) • 442
- Update Panel Fields (Online) • 170
- Update Sensitive Segment • 102
- NEXTPGM field • 199, 250, 426
 - Create/Update Screen Definition • 250
 - Update Select Fields • 199
- nonterminal definitions • 117, 371, 387, 429, 437, 439, 534
 - copying • 534
 - DGADDDing journals • 117
 - exporting • 534
 - listing • 534
 - renaming • 534
 - utilities option • 534
- NOTNULL field • 124, 332
 - Create/Update SQL Tables/TLNROWS • 124
 - Select Columns • 332
- NU field • 124, 332
 - Create/Update SQL Tables/TLNROWS • 124
 - Select Columns • 332
- NULCALL • 47
 - NULCALL • 47
- null values • 332
 - Select Columns • 332
- numeric field • 189
 - acceptable ranges of values • 189

O

- OCUST1 field • 228, 233
 - Create/Update File SEGLOOP • 233
 - Create/Update Table SEGLOOP • 228
- OCUST2 field • 228, 233
 - Create/Update File SEGLOOP • 233
 - Create/Update Table SEGLOOP • 228
- OCUST3 field • 228, 233
 - Create/Update File SEGLOOP • 233
 - Create/Update Table SEGLOOP • 228
- OEXTEND field • 207
 - Update Parameter List Extension • 207
- OF field • 170, 189, 199, 448
 - Update Batch Output Fields • 448
 - Update Output/Input/Outin Field • 170, 189

- Update Select Fields • 199
- OFFSET field • 90
 - Create/Update SSA/Command For DL/I DB/Segment • 90
- OINIT1 field • 250, 416
 - Create/Update Screen Definition • 250
- OINIT2 field • 250, 416
 - Create/Update Screen Definition • 250
- online program definition • 243
 - creating a • 243
- OP field • 77, 84, 90, 285, 335
 - Create/Update SSA/Command For DL/I DB/Segment • 90
 - Update Database Segment • 285
 - Update DBD Segment • 77
 - Update DL/I Detail Data Access • 335
 - Update DLIDSCs For Segment Member • 84
- OPCODE field • 106, 215, 291, 298, 343
 - Show/Update Data Set Default Data • 106
 - Update Data Set Record • 291
 - Update Segment Edit (SEGEDIT) • 215
 - Update SQL Detail Data Access • 298
 - Update VSAM or Sequential Detail Data Access • 343
- OPEN field • 106, 291
 - Show/Update Data Set Default Data • 106
 - Update Data Set Record • 291
- OPTION field • 90
 - Create/Update SSA/Command For DL/I DB/Segment • 90
- OPTIONS field • 170, 250, 401, 429, 462
 - Create/Update Screen Definition • 250
 - Update Panel Fields (Online) • 170
- OPTLIST field • 106, 291, 343
 - Show/Update Data Set Default Data • 106
 - Update Data Set Record • 291
 - Update VSAM or Sequential Detail Data Access • 343
- ORDERBY field • 298
 - Update SQL Detail Data Access • 298
- OS command processor • 25
 - TDF Main menu • 25
- OSEGIDX field • 228, 233
 - Create/Update File SEGLOOP • 233
 - Create/Update Table SEGLOOP • 228
- OUTATTR field • 180, 185, 189
 - Update Attribute Parameter • 180, 185
 - Update Literal Fields • 185
 - Update Output/Input/Outin Field • 189

- OUTIFIL field • 30, 394
 - Update Program Definition Defaults • 30
- outin • 30
 - field type • 30
- outin, field type • 189, 212, 215, 448
 - acceptable input values • 189, 448
 - CURSOR AT FIELD field • 212, 215
 - required • 212
 - screen field entries • 215
- OUTPUT FIELD • 281, 285, 394, 442
 - Select New Row Name • 281
 - type of editing • 442
- OUTPUT LINE field • 143
 - Specify Tables Being Joined • 143
- output mapping • 178
 - MAPOUT field • 178
- OUTTERM field • 250, 416
 - Create/Update Screen Definition • 250

P

- PAGE field • 233
 - Create/Update File SEGLOOP • 233
- PAGEKEY field • 233
 - Create/Update File SEGLOOP • 233
- PAGESAV field • 233
 - Create/Update File SEGLOOP • 233
- panel definition • 153, 156, 170, 222, 442
 - batch example • 442
 - batch, creating a • 442
 - consistency edits example • 222
 - creating a • 153
 - field characteristics • 170
- panel editing • 542, 544, 545, 547, 548, 549
 - field update rules • 542
 - general rules • 542
 - literal field rules • 544
 - long literal field rules • 549
 - rules for field with panel data • 545
 - rules for wrapped variable fields • 547
 - variable field rules • 544
 - wrapped literal fields • 548
- panel editor • 541
 - editing modes • 541
 - full screen edit mode • 541
 - line edit mode • 541
- panel image • 165, 168, 534
 - creating • 165
 - double spaced • 534
 - editing • 165

- line edit mode • 168
- line numbers • 534
- printing • 534
- panel image display window • 170, 176, 178, 180
 - Update Attribute Parameter • 180
 - Update Help Message Parameters • 176
 - Update Mapout Parameter • 178
 - Update Panel Fields (Online) • 170
- PANEL SIZE field • 55
 - Update Session Controls • 55
- panel specification • 25
 - TDF Main menu • 25
- PARENT field • 81
 - List Search Fields • 81
- PARENT/DEVICE field • 73
 - Create/Update DBD • 73
- PARENTG field • 90, 335
 - Create/Update SSA/Command For DL/I DB/Segment • 90
 - Update DL/I Detail Data Access • 335
- Parentheses • 17
 - syntax diagrams, use in • 17
- PARM LIST EXTENSION field • 189, 199, 448
 - Update Batch Output Fields • 448
 - Update Output/Input/Outin Field • 189
 - Update Select Fields • 199
- PATH field • 90, 335
 - Create/Update SSA/Command For DL/I DB/Segment • 90
 - Update DL/I Detail Data Access • 335
- PCB • 96, 102, 277
 - DBDNAME operand • 96, 277
 - DBDNAME statement • 96
 - destination of, see LTERM field • 96
 - EXPRESS value, see EXP field • 96
 - key feedback area • 96
 - KEYLEN= statement • 96
 - LTERM= statement • 96, 277
 - name to access the • 96, 102, 277
 - NAME= statement • 96
 - PROCOPT= statement • 96, 102, 277
 - PROCSEQ= statement • 96, 277
 - teleprocessing • 96, 277
 - TYPE= statement • 96
 - XFER-PCB • 277
 - XFER-PCB host variable name • 96
- PCB field • 277
 - Update DBMS Characteristics • 277
- PCB masks, DL/I • 37
 - including in program • 37
- PCB PARMS field • 77, 285
 - Update Database Segment • 285
 - Update DBD Segment • 77
- PCBNAME field • 96, 102, 215, 277
 - Create/Update PSB Or File Group • 96
 - Update DBMS Characteristics • 277
 - Update Segment Edit (SEGEDIT) • 215
 - Update Sensitive Segment • 102
- PDSCOPY DSNAME field • 55, 61
 - Color Profile Maintenance • 61
 - Update Session Controls • 55
- PF keys • 29, 53, 250
 - code to process • 250
 - how to define • 29
 - setting or changing values • 53
- PFKEYS field • 250
 - Create/Update Screen Definition • 250
- PFXLBL field • 352, 361
 - Update CICS Journal Record • 352
- PFXLTH field • 352, 361
 - Update CICS Journal Record • 352
- PGMCUST field • 30, 250, 401, 416, 429, 462
 - Create/Update Screen Definition • 250
 - Update Program Definition Defaults • 30
- PIC field • 189, 448
 - Update Batch Output Fields • 448
 - Update Output/Input/Outin Field • 189
- picture clause, defining a • 189
 - PIC field • 189
- PKYLTH field • 233
 - Create/Update File SEGLOOP • 233
- PKYUNIQ field • 233
 - Create/Update File SEGLOOP • 233
- PL/I • 30
 - specified as programming language • 30
- POSITION field • 207
 - Update Parameter List Extension • 207
- PR STORE/RENAME field • 520
 - List Panel Definitions • 520
- PRESENTATION • 514
 - Prototyping Facility menu • 514
- presentation store • 526, 529, 534
 - clear data • 529
 - clear values • 529
 - clear variable names • 529
 - editing a • 526
 - load • 529

- merge • 529
- save • 529
- presentation store editor • 551, 553, 554, 555, 556, 558, 559, 560, 561, 562, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587
 - invoking • 551
- PREVIEW field • 335, 341, 343, 356, 361
 - Update CICS Queue Data Access • 356
 - Update Database User I/O SSALIST • 341
 - Update DL/I Detail Data Access • 335
 - Update VSAM or Sequential Detail Data Access • 343
- PRINT field • 277
 - Update DBMS Characteristics • 277
- PRMSTYL • 47
 - PRMSTYL • 47
- PROCEDR field • 250, 401, 416, 429, 462
 - Create/Update Screen Definition • 250
- PROCOPT field • 96, 102, 277
 - Create/Update PSB Or File Group • 96, 102
 - Update DBMS Characteristics • 277
- PROCSEQ field • 90, 96, 277
 - Create/Update PSB Or File Group • 96
 - Create/Update SSA/Command For DL/I DB/Segment • 90
 - Update DBMS Characteristics • 277
- program • 534
 - copying • 534
 - listing • 534
 - renaming • 534
- program environment • 243, 246
 - identification • 243
- Programs • 17
 - comma • 17
- repeat symbol, use in • 17
 - parentheses • 17
- syntax diagrams, use in • 17
 - punctuation • 17
- syntax diagrams, use in • 17
- PROGTYP • 47
 - PROGTYP • 47
- prototyping commands • 529
 - View Panel Definition • 529
- prototyping facility • 25
 - TDF Main menu • 25
- PRT field • 96
 - Create/Update PSB Or File Group • 96

- PS NAME field • 514
 - Prototyping Facility menu • 514
- PSB • 65, 70, 96, 102, 534
 - creation of • 534
 - defining/maintaining a • 96
 - description of • 65, 70
 - destination for messages, see LTERM field • 96
 - importing • 65
 - last update of • 70
 - name of • 65
 - SENSEG INDICES field • 102
- PSB field • 55
 - Update Session Controls • 55
- PSBNAME field • 37, 475
 - Update Environment Definition Defaults • 37
- PSBSCHD field • 37
 - Update Environment Definition Defaults • 37

Q

- QUAL field • 120
 - DB2/Catalog Import • 120
- qualification statements • 90, 335
 - building qualification statements for request • 335
 - qualify an I/O request • 90
 - to qualify an I/O request • 335
- qualifier • 277
 - table • 277
- QUELBL field • 348
 - Update CICS Queue Record • 348
- QUELBL variable • 113
 - overriding CQNAME • 113
- queues • 65, 113, 117, 215, 267, 348, 356
 - accesses, group level variable name • 113
 - COBOL COPY, overriding • 348
 - defining • 65
 - DGADD • 267, 348
 - inheritance situation • 348
 - length, specifying • 348, 356
 - PL/I INCLUDE, overriding • 348
 - REC line, use of • 267
 - status code • 215
 - types • 348, 356
 - updating • 113, 348
 - writing to storage • 348

R

- RDBMS • 141

- Update SQL ALIAS • 141
- READSET field • 343
 - Update VSAM or Sequential Detail Data Access • 343
- RECLTH field • 106, 291, 343
 - Show/Update Data Set Default Data • 106
 - Update Data Set Record • 291
 - Update VSAM or Sequential Detail Data Access • 343
- record • 106, 291
 - I/O area • 106, 291
 - layout • 106, 291
 - maximum length • 106
- Record Identification Field length • 106
 - see KEYLEN field • 106
- record length • 343
 - maximum length • 343
- record, I/O area • 106, 291
 - overriding of • 106, 291
- REMARKS field • 250, 401, 416, 429, 462
 - Create/Update Screen Definition • 250
- RENAME field • 70, 162, 368, 523
 - List Data Administration Information • 70
 - List Panel Definitions • 162
 - List Presentation Stores • 523
- REPEAT field • 228, 233
 - Create/Update File SEGLOOP • 233
 - Create/Update Table SEGLOOP • 228
- report • 243
 - characteristics • 243
- report definition • 246
 - creating/updating/purging/showing a • 246
 - description of • 246
- REQ field • 170, 189, 442
 - Update Output/Input/Outin Field • 189
 - Update Panel Fields (Batch) • 442
 - Update Panel Fields (Online) • 170
- REQUEST field • 267
 - Create/Update Data Group • 267
- RESLTCC field • 298
 - SPBROWSE, SPTRNACT and SPRDNEXT Data Access • 298
- RESLTNR KEY field • 298
 - SPBROWSE, SPTRNACT and SPRDNEXT Data Access • 298
- RESLTPR field • 298
 - SPBROWSE, SPTRNACT and SPRDNEXT Data Access • 298
- RESULTS • 47

- RESULTS • 47
- REUSE field • 106
 - Show/Update Data Set Default Data • 106
- RID • 291
 - length of • 291
- RID field length • 106
 - RID field length • 106
- RIDFLD • 267
 - WORKSPA field • 267
- RMNAME field • 73
 - Create/Update DBD • 73
- ROW NAME • 132
 - Create/Update IDMS SQL Tables/TLNROWS • 132
- ROW NAME field • 332
 - Select Columns • 332

S

- SAVEKEY field • 228, 233
 - Create/Update File SEGLOOP • 233
 - Create/Update Table SEGLOOP • 228
- SCHEMA • 47, 132, 141
 - Create/Update IDMS SQL Tables/TLNROWS • 132
 - SCHEMA • 47
 - Update SQL ALIAS • 141
- SCHFLDC field • 233
 - Create/Update File SEGLOOP • 233
- SCHFLDI field • 233
 - Create/Update File SEGLOOP • 233
- SCHFLDL field • 233
 - Create/Update File SEGLOOP • 233
- SCONSIS field • 199, 426
 - Update Select Fields • 199
- SCOPE field • 448
 - Update Batch Output Fields • 448
- screen • 150, 243
 - Add/Update SQL Join Alias • 150
 - characteristics • 243
 - screen definition • 117, 246
 - creating/updating/purging/showing a • 246
 - description of • 246
 - DGADding journals • 117
 - screen field • 215
 - entries • 215
 - screen field, entries • 212
 - required • 212
 - screen program IDs • 25
 - F100 • 25

SCREEN XFEDIT/SEGEDIT field • 250

 Create/Update Screen Definition • 250

screens • 29, 30, 36, 47, 53, 55, 65, 70, 73, 77, 81, 84, 90, 95, 96, 102, 104, 106, 113, 117, 120, 122, 124, 132, 141, 143, 145, 147, 156, 162, 165, 168, 170, 176, 178, 180, 185, 189, 199, 207, 209, 212, 215, 228, 233, 246, 250, 267, 277, 281, 285, 291, 298, 309, 312, 316, 320, 323, 327, 328, 332, 335, 341, 343, 348, 352, 356, 361, 365, 368, 371, 373, 382, 385, 387, 394, 401, 410, 416, 423, 426, 429, 437, 439, 442, 446, 454, 458, 462, 475, 479, 481, 483, 486, 493, 514, 520, 523, 526, 529

 accessing Show/Purge • 73

 Catalog/Import DB2 Tables • 120

 Create/Update Data Group • 267

 Create/Update DBD • 73

 Create/Update File SEGLOOP • 233

 Create/Update IDMS SQL Tables/TLNROWS • 132

 Create/Update PSB or File Group • 96

 Create/Update Screen Definition • 250

 Create/Update SQL Tables/TLNROWS • 124

 Create/Update SSA/Command For DL/I DB/Segment • 90

 Create/Update Table SEGLOOP • 228

 Data Administration menu • 65

 Edit Panel Image • 165

 FETCH Details • 323

 Field Extension • 327

 GET DIAGNOSTICS Combined Information • 320

 GET DIAGNOSTICS Condition Information • 316

 GET DIAGNOSTICS List Screen • 309

 GET DIAGNOSTICS Statement Information • 312

 Line Edit • 168

 List Consistency Edits • 209

 List Data Administration Information • 70

 List Panel Definitions • 162, 520

 List Presentation Stores • 523

 List Search Fields • 81

 List SQL Tables/Joins • 122

 Online Program Definition menu • 246

 Panel Definition menu • 156

 Prototyping Facility menu • 514

 Select Columns • 332

 Select New Row Name • 281

 Show/Purge PSBs Or File Groups • 96

 Show/Purge Screen Definition • 250

 Show/Update Data Set Default Data • 106

 Specify Tables Being Joined • 143

 Update Attribute Parameter • 180

 Update Batch Literal Fields • 446

 Update CICS Journal Default Data • 117

 Update CICS Journal Record • 352

 Update CICS Queue Data Access • 356

 Update CICS Queue Default Data • 113

 Update CICS Queue Record • 348

 Update Cross-Field Edit (XFEDIT) • 212

 Update Data Set Record • 291

 Update Database Segment • 285

 Update Database User I/O SSALIST • 341

 Update DBD Segment • 77

 Update DBMS Characteristics • 277

 Update DL/I Detail Data Access • 335

 Update DLIDSCs For Segment Member • 84

 Update Environment Definition Defaults • 36

 Update Help Message Parameters • 176

 Update Literal Fields • 185

 Update Mapout Parameter • 178

 Update Output/Input/Outin Field • 189

 Update Panel Fields (Batch) • 442

 Update Panel Fields (Online) • 170

 Update Panel Group • 454

 Update Parameter List Extension • 207

 Update PF Keys Definition • 53

 Update Program Definition Defaults • 30

 Update Segment Edit (SEGEDIT) • 215

 Update Select Fields • 199

 Update Sensitive Segment • 102

 Update Sensitive TLNROWS • 104

 Update Session Controls • 55

 Update SQL ALIAS • 141

 Update SQL Detail Data Access • 298

 Update SQL Join? Access Columns • 147

 Update SQL Join? Join Columns • 145

 Update Stored Procedure Environment Definition Defaults • 47

 Update User-Defined Datatypes • 328

 Update VSAM or Sequential Detail Data Access • 343

 User Profile Maintenance • 29

 Verify DLIDSC Count • 95

 View Panel Definition • 529

 View Presentation Store • 526

search field • 81

- length of • 81
- picture clause • 81
- starting position • 81
- type of data • 81
- SEARCH FIELDS field • 77
 - Update DBD Segment • 77
- SECTION field • 250, 401, 416, 429, 462
 - Create/Update Screen Definition • 250
- SEGEDIT • 209, 215, 222
 - consistency edits example • 222
 - copying an existing • 215
 - List Consistency Edits • 209
- SEGKEY field • 343
 - Update VSAM or Sequential Detail Data Access • 343
- SEGLOOP • 170, 212, 215, 228, 233
 - Create/Update File SEGLOOP • 233
 - create/update table SEGLOOP • 228
 - first field name • 228, 233
 - last field name • 228, 233
 - range of, see SEGLOOP DELIMITERS field • 170
 - SEGLOOP • 212, 215
 - type of • 228
- SEGLOOP DELIMITERS field • 170
 - Update Panel Fields (Online) • 170
- SEGLOOP field • 156, 228, 233
 - Create/Update File SEGLOOP • 233
 - Create/Update Table SEGLOOP • 228
 - Panel Definition menu • 156
- SEGLTH field • 106, 291
 - overriding the • 106, 291
- SEGM field • 77, 285
 - Update Database Segment • 285
 - Update DBD Segment • 77
- segment • 73, 77, 84, 90, 95, 199, 267, 285
 - adding to a DBD • 267
 - I/O area of • 77, 285
 - layout • 77, 285
 - maintaining position • 90
 - name of • 84, 90, 95
 - name of, see NAME field • 73
 - parent segment offset • 90
 - requested for Auto I/O • 199
 - retrieving a key • 77
 - variable length • 90
- SEGMENT field • 77, 84, 90, 95, 285, 335, 341
 - Create/Update SSA/Command For DL/I DB/Segment • 90
 - Update Database Segment • 285
 - Update Database User I/O SSALIST • 341
 - Update DBD Segment • 77
 - Update DL/I Detail Data Access • 335
 - Update DLIDSCs For Segment Member • 84
 - Verify DLIDSC Count • 95
- segment key • 73, 81
 - name of • 73
 - number of bytes • 73, 81
 - starting byte position • 73, 81
- SEGMENT KEY field • 73
 - Create/Update DBD • 73
- SEGMENT NAME field • 215
 - Update Segment Edit (SEGEDIT) • 215
- segment qualification option • 84, 90
 - specifying the • 84, 90
- segment, I/O area • 77, 285
 - overriding of • 77, 285
- select • 30
 - field type • 30
- select field • 199, 281, 332, 462
 - J-100 paragraph • 199
 - Select Columns • 332
 - Select New Row Name • 281
- SELECT FIELDS field • 250
 - Create/Update Screen Definition • 250
- select, field type • 189, 212, 215, 448
 - acceptable input values • 189, 448
- CURSOR AT FIELD field • 212, 215
 - required • 212
 - screen field entries • 215
- SELKEY field • 199
 - Update Select Fields • 199
- SENCOLS field • 298
 - Update SQL Detail Data Access • 298
- SENSEG statement, PROCOPT= • 102
 - see PROCOPT field • 102
- SEQ field • 73, 96, 102, 104, 209, 479, 486
 - Create/Update DBD • 73
 - Create/Update PSB Or File Group • 96
 - List Consistency Edits • 209
 - Update Sensitive Segment • 102
 - Update Sensitive TLNROWS • 104
- sequence number • 81
 - List Search Fields screen • 81
- sequence number field • 77
 - Update DBD Segment • 77
- sequential • 106, 291
 - ACCESS field • 106, 291

- sequential file • 65, 70
 - description of • 65, 70
 - last update of • 70
 - name of • 65
- session controls • 29, 30
 - User Profile Maintenance • 29
- SET field • 356
 - Update CICS Queue Data Access • 356
- SHIFT field • 534
 - Utilities menu • 534
- SIZE field • 156, 250, 416, 426, 429, 442, 462, 479, 483, 486
 - Create/Update Screen Definition • 250
 - Panel Definition menu • 156
 - Update Panel Fields (Batch) • 442
- SKIPAFD field • 454
 - Update Panel Group • 454
- SKIPBEF field • 454
 - Update Panel Group • 454
- source code • 534
 - exporting program definition • 534
- SPACMPT field • 40, 373
 - Update Environment Definition Defaults • 40
- SPASIZE field • 37, 40, 373, 410
 - Update Environment Definition Defaults • 37, 40
- SPASTG field • 37
 - Update Environment Definition Defaults • 37
- SPEC field • 189, 199, 448
 - Update Batch Output Fields • 448
 - Update Output/Input/Outin Field • 189
 - Update Select Fields • 199
- SPECIAL EDITS • 513
 - use with prototyping • 513
- SQL • 122, 124, 215
 - error condition return codes • 215
 - key field and variables correspondence • 215
 - qualifier • 122, 124
- SQL QUAL.NAME field • 122
 - List SQL Tables/Joins • 122
- SQLMOD • 47
 - SQLMOD • 47
- SRC • 209
 - List Consistency Edits • 209
- SRCHFLD NAME field • 81
 - List Search Fields • 81
- SSA • 81, 335, 625, 626, 627, 628, 629, 630, 631, 632
 - building SSA statements for request • 335

- call against the data base • 335
- IMS key picture clause • 81
- KEYPIC field • 81
 - search field picture clause • 81
- SSA command code field • 84, 90, 285, 335
 - overriding of • 84, 285
 - overriding the • 90, 335
- SSA LIST HOST VARIABLE field • 341
 - Update Database User I/O SSALIST • 341
- SSALIST field • 215, 335, 341
 - Update Database User I/O SSALIST • 341
 - Update DL/I Detail Data Access • 335
 - Update Segment Edit (SEGEDIT) • 215
- SSAs • 285, 335
 - listing • 285, 335
- START field • 73, 81, 486
 - Create/Update DBD • 73
 - List Search Fields • 81
- statement • 55
 - compression of • 55
 - repeating a • 55
- STAYRES • 47
 - STAYRES • 47
- STBRKEY field • 233
 - Create/Update File SEGLOOP • 233
- SUB field • 526
 - View Presentation Store • 526
- SUMMARY • 442, 454
 - GPTYPE field • 442, 454
- SWAP command • 513, 514
 - prototyping facility restriction • 513
- SYNONYM • 132
 - Create/Update IDMS SQL Tables/TLNROWS • 132
- SYNONYM field • 124, 277
 - Create/Update SQL Tables/TLNROWS • 124
 - Update DBMS Characteristics • 277
- Syntax diagrams • 17, 25
 - reading (how to) • 17, 25
- SYSID field • 113, 348, 356
 - Update CICS Queue Data Access • 356
 - Update CICS Queue Default Data • 113
 - Update CICS Queue Record • 348
- system log • 117
 - CICS reserved ID • 117

T

- table • 65, 120, 122, 124, 143, 145, 147, 150, 277, 281, 332

- Add/Update SQL Join Alias • 150
- correlation name • 143, 145, 147
- description of • 65
- importing a • 120
- key name • 145
- name of • 65, 120, 122
- naming a • 120, 124, 143, 147, 332
- qualifier • 277
- row • 281
- TABLE field • 277
 - Update DBMS Characteristics • 277
- TABLE NAME field • 143, 281, 332
 - Select Columns • 332
 - Select New Row Name • 281
 - Specify Tables Being Joined • 143
- TABlename field • 120
 - Catalog/Import DB2 Tables • 120
- TBLNAME field • 104
 - Update Sensitive TLNROWS • 104
- temporary storage
 - DGADD • 267
 - handle conditions • 356
 - queue names • 113, 348, 356
 - status codes • 215
- TEXT field • 185, 446
 - Update Batch Literal Fields • 446
 - Update Literal Fields • 185
- TLN field • 120
 - Catalog/Import DB2 Tables • 120
- TLNNAME • 132, 141, 150
 - Add/Update SQL Join Alias • 150
 - Create/Update IDMS SQL Tables/TLNROWS • 132
 - Update SQL ALIAS • 141
- TLNNAME field • 65, 120, 122, 124, 143, 145, 147, 277
 - Catalog/Import DB2 Tables • 120
 - Create/Update SQL Tables/TLNROWS • 124
 - List SQL Tables/Joins • 122
 - max size • 65
 - Specify Tables Being Joined • 143
 - Update DBMS Characteristics • 277
 - Update SQL Join? Access Columns • 147
 - Update SQL Join? Join Columns • 145
- TLNROW • 141
 - Update SQL ALIAS • 141
- TLNROW field • 104, 281
 - Select New Row Name • 281
 - Update Sensitive TLNROWS • 104
- TMPCOMT field • 281
 - Select New Row Name • 281
- TMPNAME field • 281
 - Select New Row Name • 281
- TMPQUAL field • 281
 - Select New Row Name • 281
- TO field • 156
 - Panel Definition menu • 156
- TO HEADER field • 534
 - Utilities menu • 534
- TO ID field • 534
 - Utilities menu • 534
- TOPDTL • 442, 454
 - GPTYPE field • 442, 454
- TOPPAGE • 442, 454
 - GPTYPE field • 442, 454
- TOTREF field • 448
 - Update Batch Output Fields • 448
- TOTSIZE field • 448
 - Update Batch Output Fields • 448
- TPBSTG field • 37
 - Update Environment Definition Defaults • 37
- TRACE field • 37, 40, 45, 373, 387, 410, 423, 475
 - Update Environment Definition Defaults • 37, 40, 45
- TRANCDE field • 40, 373, 387, 410
 - Update Environment Definition Defaults • 40
- TRANFLD field • 40, 373
 - Update Environment Definition Defaults • 40
- TRANMFS field • 40, 373
 - Update Environment Definition Defaults • 40
- transfer work area • 30, 199, 394
 - field name for mapping data • 199
 - including COPY member names • 30
- transferring control from panel to panel • 529
 - View Panel Definition • 529
- transient data • 113, 215, 267, 348, 356
 - DGADD • 267
 - handle conditions • 356
 - queue names • 113, 348, 356
 - status codes • 215
- translate lower case to upper case • 514
 - prototyping facility • 514
- TSO • 44, 55
 - environment • 55
 - environment parameters • 44
- Type • 81, 132, 458

- Create/Update IDMS SQL Tables/TLNROWS • 132
- List Search Fields screen • 81
- TYPE (TLNROW) field • 124
 - Create/Update SQL Tables/TLNROWS • 124
- TYPE field • 73, 81, 96, 106, 113, 124, 209, 277, 291, 332, 348, 356, 382, 454, 534
 - Create/Update DBD • 73
 - Create/Update PSB Or File Group • 96
 - Create/Update SQL Tables/TLNROWS • 124
 - List Consistency Edits • 209
 - List Search Fields • 81
 - Select Columns • 332
 - Show/Update Data Set Default Data • 106
 - Update CICS Queue Data Access • 356
 - Update CICS Queue Default Data • 113
 - Update CICS Queue Record • 348
 - Update Data Set Record • 291
 - Update DBMS Characteristics • 277
 - Update Panel Group • 454
 - Utilities menu • 534

U

- U field • 90, 442
 - Create/Update SSA/Command For DL/I DB/Segment • 90
 - Update Panel Fields (Batch) • 442
- U-100 section • 30
 - U-100-request-segment-name program section • 30
 - U-100-USER-IO program section • 30
- Update Cross-Field Edit (XFEDIT) • 222
 - consistency edits example • 222
- UPDATE DBD field • 77
 - Update DBD Segment • 77
- UPDATE ENVIRON DEF DEFAULTS field • 30
 - Update Program Definition Defaults • 30
- UPDATE field • 70, 77, 120, 162, 298, 520, 523
 - Catalog/Import DB2 Tables • 120
 - List Data Administration Information • 70
 - List Panel Definitions • 162, 520
 - List Presentation Stores • 523
 - Update DBD Segment • 77
 - Update SQL Detail Data Access • 298
- Update Segment Edit (SEGEDIT) • 222, 228
 - consistency edits example • 222
- UPDCOLS field • 327, 328
 - Field Extension • 327
- UPPER CASE LITERALS field • 55

- Update Session Controls • 55
- USAGE field • 185, 189, 199, 281, 285, 291, 348, 352, 446, 448
 - Select New Row Name • 281
- Update Batch Literal Fields • 446
- Update Batch Output Fields • 448
- Update CICS Journal Record • 352
- Update CICS Queue Record • 348
- Update Data Set Record • 291
- Update Database Segment • 285
- Update Literal Fields • 185
- Update Output/Input/Outin Field • 189
- Update Select Fields • 199
- USE field • 170, 176, 178, 180, 442
 - Update Attribute Parameter • 180
 - Update Help Message Parameters • 176
 - Update Mapout Parameter • 178
 - Update Panel Fields (Batch) • 442
 - Update Panel Fields (Online) • 170
- USECNT field • 84
 - Update DLIDSCs For Segment Member • 84
- USECOL1 field • 534
 - Utilities menu • 534
- USED BY field • 95
 - Verify DLIDSC Count • 95
- user exec requests • 267
 - online programs • 267
- USER field • 70, 162, 368, 520, 523
 - List Data Administration Information • 70
 - List Panel Definitions • 162, 520
 - List Presentation Stores • 523
- User I/O • 298
 - GROUPBY clause • 298
 - HAVING clause • 298
 - ORDERBY clause, see ORDERBY field • 298
- USER MODE field • 55
 - Update Session Controls • 55
- user profile maintenance • 25
 - TDF Main menu • 25
- USEREXEC • 113, 117
 - replace CICS journal name • 117
 - replace CICS queue name • 113
- USEREXEC requests • 267
 - online programs • 267
- USERID field • 120
 - Catalog/Import DB2 Tables • 120
- USGCOP1 field • 37
 - Update Environment Definition Defaults • 37
- USGCOP2 field • 37

- Update Environment Definition Defaults • 37
- USGCOPY field • 37, 40, 44, 45, 373, 387, 410, 423, 475
- Update Environment Definition Defaults • 37, 40, 44, 45
- utilities • 25, 29
- TDF Main menu • 25

V

- VARLTH field • 90
- Create/Update SSA/Command For DL/I DB/Segment • 90
- view • 120
- importing a • 120
- VSAM • 106, 215, 291
- ACCESS field • 106, 291
- data set name, accessed by the SEGEDIT • 215
- direct access • 106, 291
- dynamic access • 106, 291
- ESDS • 106, 291
- KSDS • 106, 291
- random access • 106, 291
- RRDS • 106, 291
- sequential access • 106, 291
- VSAM SEGMENT LTH field • 215
- Update Segment Edit (SEGEDIT) • 215

W

- WHEN field • 215
- Update Segment Edit (SEGEDIT) • 215
- WHERE field • 267, 281, 298
- Create/Update Data Group • 267
- Select New Row Name • 281
- Update SQL Detail Data Access • 298
- WKAREA field • 250, 401, 416, 429, 462
- Create/Update Screen Definition • 250
- WKSPAIN field • 40, 373, 410
- Update Environment Definition Defaults • 40
- WKSPAIO field • 40, 373, 410
- Update Environment Definition Defaults • 40
- WKSPASZ field • 40, 373, 410
- Update Environment Definition Defaults • 40
- WLMENV • 47
- WLMENV • 47
- working storage • 199
- field name for mapping data • 199
- WORKSPA • 40, 373
- database • 40, 373

- database initialization code • 40, 373
- database, size of • 40
- retrieval and writing of WORKSPA databases • 40, 373

X

- XFEDIT • 212, 222
- copying an existing • 212
- example • 222
- XFER-PCB host variable name • 277
- ABCALL field • 277
- MSGCALL field • 277
- XFERWKA field • 30, 250, 401, 416, 429
- Create/Update Screen Definition • 250
- Update Program Definition Defaults • 30
- XFIELD • 209
- List Consistency Edits • 209