CA Telon® Application Generator

Utilities Guide

r5.1

CA Product References

This document references the following CA products:

- CA Telon®
- CA IDMS SQL

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 available on the CA support website at http://ca.com/support.

Contents

Chapter 5: Import/Export Procedures 55

Chapter 6: Automated Documentation 75

Chapter 1: Introduction

This guide describes how to use the utilities provided with the CA Telon Application Generator, formerly known as CA Telon. In this guide, the product is referred to simply as CA Telon. The CA Telon utilities described in this guide perform the following functions:

- Transport data between the TDF and a transport file
- Export TDF definitions to a source library
- Import CA Telon source into the TDF
- Generate CA Telon Automated Documentation reports

Also included in this guide are Transport installation procedures, Transport error codes and messages, considerations for a multiple TDF environment, and object and record relation charts.

System Overview

This chapter provides an overview of the CA Telon Transport system. As you work through details of the system in later chapters, refer to this chapter when you need to see how the system components function together.

This chapter covers the following subjects:

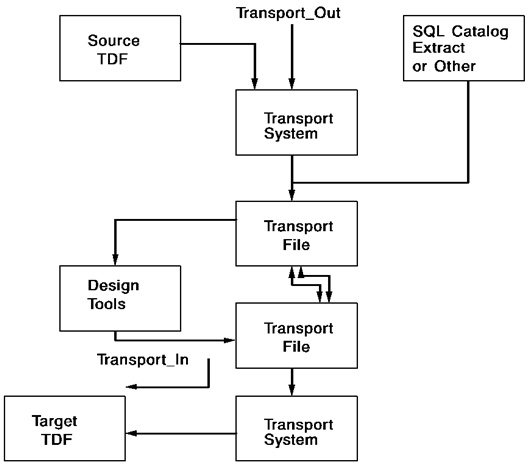
- System flow
- Objects
- System components

Audience

Personnel responsible for using CA Telon Utilities should use this guide. It assumes a familiarity with basic CA Telon terminology (for example, Maximum Severity Level, Presentation Stores), and basic computer and database terminology (for example, DB2, SQL, IMS, JCL).

System Flow

The primary purpose of the CA Telon Transport system is to move data to a TDF. Normally, the data comes from another TDF, but SQL Catalog Extract systems or other utilities can also generate it.



Terminology

The previous figure uses the following terminology:

- Source TDF—Specifies the originating TDF from which you want to move TDF information.
- Target TDF—Specifies the destination TDF to which you want to move the TDF information.
- SQL Catalog Extract or Other—Runs a job on the CA Telon or another source platform to create the transport file, or a user-written program that creates the transport file.
- Transport File—Specifies a standard ASCII or EBCDIC file used to store and transport TDF information.
- Transport System—Contains a driver program and multiple application programs. It is the vehicle to move data between the TDF and the transport file.
- Transport-Out—Moves data from the source TDF to the transport file.
- Transport-In—Moves TDF data from a transport file to the target TDF.
- Design Tools—Uses front-end design facilities of other vendors to create transport files of information to move to target TDFs.

Objects

The CA Telon Design Facility (TDF) contains different types of information. The Transport system refers to each distinct type of information as a CA Telon object.

User-Defined and Internal Objects

There are two types of CA Telon objects: user-defined and internal. User-defined objects are collections of information that a TDF user supplies to define some part of a CA Telon application program, Data Administration object (for example, IMS DBD, SQL Table, File, CICSJRNL, PSB, or File Group) or presentation store. Internal objects control and manage the TDF, such as system and user profiles and defaults. The CA Telon system administrator maintains internal objects.

Root and Dependent Objects

There are two levels of CA Telon objects:

- Root objects—Are main objects in Transport. Each root object has object-related information associated with it and can have one or more dependent objects associated with it.
- Dependent objects—Are components of root objects that structure and further define the contents of the root objects.

User-defined root objects of the TDF

The following table displays the user-defined root objects of the TDF;

Name	Type of Information
PANEL	Online screen, Report/Batch/CICS Nonterminal report, IMS Report Definition report Definitions (PI, PD)
PROGRAM	Online/Batch/Driver/Report/CICS Nonterminal/Stored Procedure Program Definitions (SD, BD, DR, RD, ND, SP)
IMSDBD	IMS Database Definitions
TABLE	SQL Table Definitions
JOIN	Table Join Definitions

Name	Type of Information
FILE	VSAM/Sequential File Definitions
CICSQUE	CICS Queue Definitions
CICSJRNL	CICS Journal Definitions
PSB	IMS PSB Definitions
FILEGRP	File Group Definitions
PSTORE	Presentation Stores

Internal root objects of the TDF

The following table displays the user-defined root objects of the TDF;

Name	Type of Information
USERPROF	User Profile Definitions
INSTALL	Installation Defaults and Security Definition

Relationships Between Objects

You can view the Transport system as using a pseudo-entity relationship (E-R) model to encode TDF information. The relationship between objects is an implied Contains Relationship without attributes. For example, the PANEL root object has a GROUP dependent object. An E-R model would show this relationship as a PANEL entity and a GROUP entity with a PANEL_Contains_GROUP relationship connecting the two entities.

You can also look at the relationships between root and dependent objects as a hierarchical model, with the root object as an IMS root segment and the dependent object(s) as its children. If you use this model, some objects could be implemented as physical databases and others as logical databases.

This guide uses the root object/dependent object terminology. This standardized terminology avoids the confusion which would result from overlapping terms.

Note: In the remainder of this guide the term *object* refers to a root object and all of its dependent objects.

Transport-Out and Transport-In

Transport-Out and Transport-In are the basic functions (or operations) in Transport. Transport-Out takes one or more objects from the source TDF, encodes them in Transport object format, and writes them to a Transport file. Transport-In takes a Transport file created by Transport-Out, another vendor, or a user application, and loads the objects found on the file into the target TDF.

System Components

The next diagram illustrates the major components of the CA Telon Transport system.

Component descriptions

The following components make up the CA Telon Transport system:

- TDF Database—Stores objects. The source TDF is the place from which you want to move objects. The target TDF is the place where you want to move the objects.
- Transport System—Contains a driver program and one or more application programs for each object defined. It is the vehicle for moving TDF data.
- Transport File (TPTRAN)—Specifies the standard ASCII/EBCDIC file used either as input to or output from the Transport system.
- Transport control file (TPCTL)—Controls Transport system execution by using control cards. The transport control file contains a minimum of two control cards: a TRANPORT control card and an END card. You can also supply object control cards to selectively control the transport of specific root objects or groups of root objects.
- Transport Status Report (TPRPT)—Reports the actions taken by the Transport system and any detected errors.

Chapter 2: Transport File

The transport file is a sequential 80-byte ASCII or EBCDIC file that contains encoded TDF information. Most transport records use a standard identifier-data format. However, within the CUSTCODE dependent object in the PROGRAM root object, unformatted transport records can define procedural code and documentation for a program.

This chapter describes the format and structure of the transport file and covers the following subjects:

- Identifiers
- Transport record processing considerations
- Sample transport file

Identifiers

The identifier is a one-to eight-byte uppercase character name (starting in column one) that identifies the transport record. The data value associated with the record identifier begins in column ten of the transport record.

The Transport system recognizes the identifiers listed in this table:

Record	Identifier
Transport Header Record	TRANPORT
Root Object Header Record	Root Object Name
Dependent Object Header Record	Dependent Object Name
Object-Related Transport Record	CA Telon Statements and their parameters
Trailer Record	END (for Root or Dependent Objects)
Comment Record	*
Other	Described later in this section

Transport Header Record

The transport file begins with a transport header record that has the following format:

```
TRANPORT mm/dd/yy hh:mm:ss user-name

release.version
```

The table below describes each field in the transport header record.

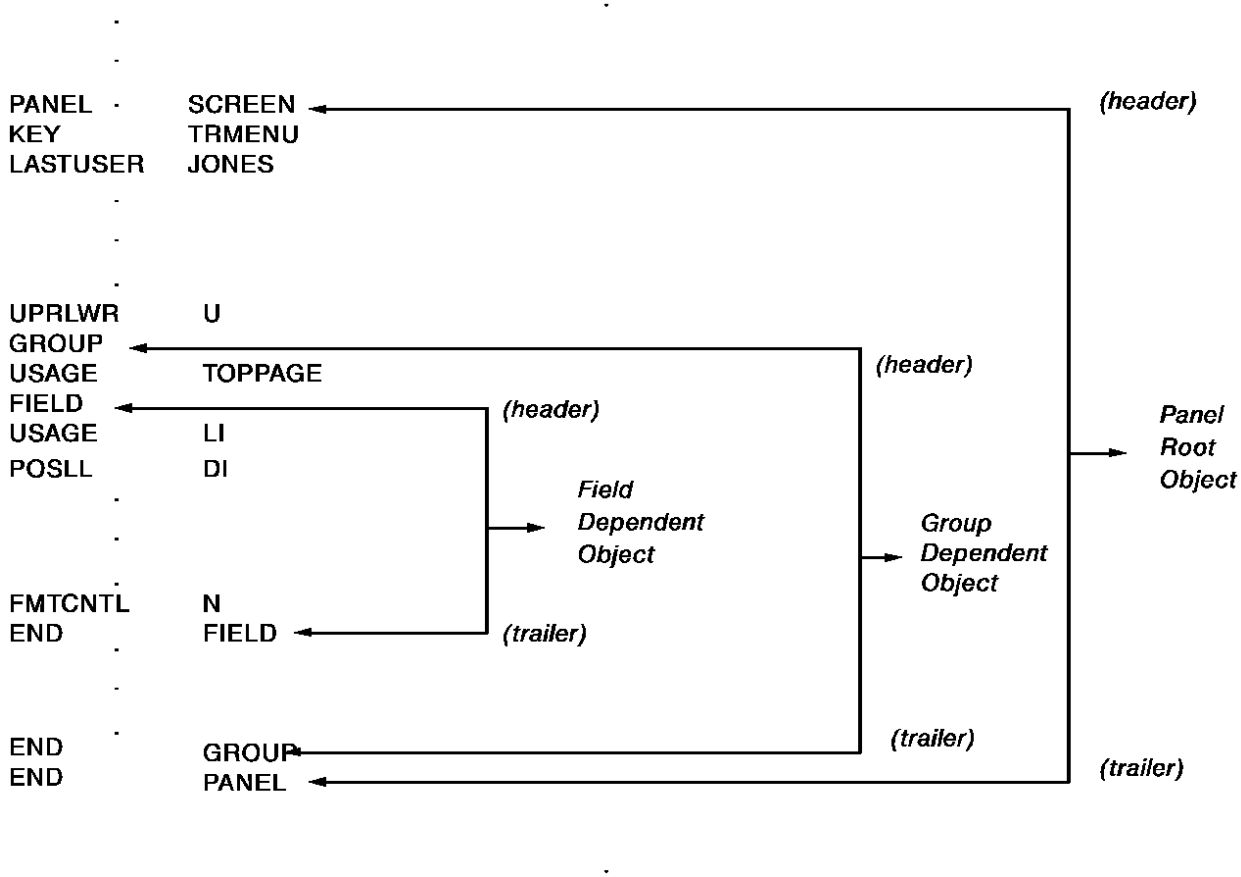
Field	Column	Length	Contents
Identifier	1-9	8	TRANPORT
Date	10-17	8	Date transport file created in *MM/DD/YY* format
Time	18-25	8	Time transport file created in *HH:MM:SS* format
User name	26-65	40	Name of the user who created the transport file
Release level	66-68	3	CA Telon release level used to create the transport file. The release level is specified as *release.version*; for example, 5.1.

Object Header and Trailer Names

A root object header transport record and a trailer transport record delimit each root object on the transport file. In between are the transport records for the object's information, followed by transport records of any dependent object. Header and trailer records also delimit each dependent object. The next figure shows header and trailer positioning for root and dependent objects.

The identifiers used for root objects are the names identified in the previous illustration. The identifier used for an object's Trailer is END with the root object or dependent object name in column ten. All root object headers are unique within the transport system. Dependent objects are unique within a root object.

Root object and dependent object transport records usually have data values associated with them. If so, the values are either keys or type identifiers.



The identifiers used for root objects are the names identified in the previous illustration. The identifier used for the Trailer of an object is END with the root object or dependent object name in column ten. All root object headers are unique within the transport system. Dependent objects are unique within a root object.

Root object and dependent object transport records usually have data values associated with them. If so, the values are either keys or type identifiers.

CA Telon Statement/Parameter Identifiers

The identifiers for the object-related transport records are usually CA Telon statement or parameters. For more information about statements and their parameters, see the *Programming Concepts Guide*.

Comment (*)

To identify a comment transport record, an asterisk (*) in column 1 is used as the identifier.

For more information with a detailed description of comment records, see the appendix "Transport File Description Charts."

Other Identifiers

There are additional transport records that do not appear as CA Telon statement or parameters because these records are only used by the TDF. Whenever possible for these transport records, the transport record identifiers are usually the same as the name of the associated field literal on the TDF screens. The data value associated with the transport record is the maximum length used in the TDF, and its contents are the same as entered into the TDF.

For transport records associated with one or more CA Telon statements or parameters, the data value associated with the record is for the length defined in the transport record chart. This is usually the same value as the associated statement or parameter. The data contents or format is defined by the CA Telon statement or parameter that is associated with the transport record. This is also the same format as that used to enter the data into the TDF.

The transport reference chart identifies the CA Telon statement or parameter (or TDF screens) that are associated with each transport record.

Note: Some CA Telon statement/parameters are greater than the 71 bytes of data that can fit on the remainder of a transport record. For these parameters, you can use multiple transport records. Each transport record can obtain a segment of the parameters where the transport record chart defines the length of each segment. This length is usually the same value as the size used for entry on the TDF.

Transport Record Processing Considerations

Important transport record processing considerations are:

- When to transport records
- Order for transporting records

When to Transport Records

When Transport-out creates a file, that file contains records for all dependent objects which contain existing data. If a transport record is alphanumeric and contains spaces or is numeric and contains zeros, no transport record is required. Transport files created by an outside source must do the same. Transport files from non-CA Telon sources that do not conform to CA Telon standards will not be transported into a TDF.

Order for Transporting Records

You can transport objects in any sequence on the transport file. Within a root or dependent object, the header must be first. Required transport records are next, and any optional transport records can be interspersed. Any related dependent objects and their associated transport records follow. The root object and dependent object is completed with a Trailer transport record.

Dependent objects can be nested. The order within root objects is shown next.

- Root Object Header
- Required Transport Records
- (Optional Transport Records)
- (Related Dependent Objects and Records)
- Root Object Trailer

The order within dependent objects is shown next.

- Dependent Object Header
- Required Transport Records
- (Optional Transport Records)
- (Related Dependent Objects and Records)
- Dependent Object Trailer

Sample Transport File

The next example displays a sample Transport file for a PANEL object of the Training System menu screen. It includes:

- The Transport record type in the left column:
  - H (Transport header)
  - C (Comment record)
  - R (Root object)
  - D (Dependent object)
  - P (Parameter record)
  - T (Trailer record)
- The Transport identifier
- The data value

```
  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--000100
01
H TRANPORT 01/26/0209:32:57SAMPLE PROD DEV TRANSPORT FILE     x.x  00020001
C *    FOR TRANSPORT USE ONLY. DO NOT USE FOR ARCHIVE.        00020002
R PANEL  SCREEN                              00030001
P KEY   TRMENU                              00040001
P DESC   PANEL FOR SAMPLE TRANSPORT FILE              00050001
P LASTUSER JOANNET                            00060001
P LASTUPD 010126                             00070001
P UPDTIME 0931                               00080001
P CRTDATE 010126                              00090001
P LASTUSED 010126                              00100001
P INPCHAR <                                 00110001
P OUTCHAR >                                 00120001
P OICHAR  +                                 00130001
P SELCHAR |                                 00140001
P LITBRK  \                                 00150001
P SIZELL  024                                00160001
P SIZECCC 080                                00170001
P UPRLWR  U                                 00180001
D GROUP                                     00190001
P USAGE  TOPPAGE                               00200001
D FIELD                                     00210001
P USAGE  LI                                  00220001
P POSLL  01                                  00230001
P POSCCC  022                                 00240001
P LTH   020                                 00250001
P TEXT   EMPLOYEE SYSTEM MENU                         00260001
P FMTCNTL N                                  00270001
T END   FIELD                                00280001
T END   GROUP                                00290001
T END   PANEL                                00300001
```

Chapter 3: Transport Control File

The execution of the Transport system is managed by using control cards within the Transport control file. The following sections describe the Transport control file and how to use it:

- Basic file contents
- Control card descriptions
- System processing using control cards

File Contents

The Transport control file tells the Transport system what to do. There are three types of control cards in this file:

- TRANSPORT—Supplies global information for transport processing
- Object control cards—Provide data about the objects to be transported
- END—Terminates transport processing

The next figure represents the Transport control file schematically:

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7--00010000
TRANPORT XXXXXXYYYZZZZZZZU(40)                                        00020000
TRANPORT IMPORTSELIGNORE SMPLE PROD DEV TRANSPORT FILE                 00030000
                                                                       00040000
NNNNNNNN KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKK PPPPPPPSS          00050000
TABLE    TELON.TRGEMPL                              MERGE  04          00060000
TABLE    TELON.TRGEMPL1                             MERGE  04          00070000
                                                                       00070100
NNNNNNNN KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKK PPPPPPPSS          00070200
IMSDBD   TRGDBDV1                                   MERGE  04          00070300
IMSDBD   TRGDBDVX                                   MERGE  04          00070400
                                                                       00070500
NNNNNNNN KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKK PPPPPPPSS          00070600
FILE     TRGEMPLV                                   MERGE  04          00070700
FILE     TRGEMPLA                                   MERGE  04          00070800
FILE     EMPMAST                                    MERGE  04          00070900
NNNNNNNN KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKK PPPPPPPSS          00071000
CICSQUE  EMPLQ                                      MERGE  04          00072000
CICSJRNL TNJOURN1                                   MERGE  04          00073000
                                                                       00074000
END                                                                    00075000
```

Each of the control cards within the Transport control file are described next.

TRANPORT Control Card

The format of the TRANPORT control card is shown next.

```
----+----1----+----2----+----3----+----4----+----5----+----6-----
TRANPORT xxxxxxyyyzzzzzzzu(40)
```

The fields on the TRANPORT control card are as follows:

TRANSPORT

Columns 1-8. The TRANPORT control card identifies processing that the system is to perform.

One TRANPORT control card is required for a transport process. For readability, it should be the first control card in sequence. For processing, it must appear in sequence before the END control card.

xxxxxx

Columns 10-15. Identifies the transport process.

Values are as follows:

- EXPORT—Transport one or more objects out of a TDF
- IMPORT—Transport one or more objects into a TDF

yyy

Columns 16-18. Specifies the selection of user-defined objects to be transported.

Valid values are as follows:

- ALL(*Defalt*)—Transport all user-defined objects from or into the TDF
- SEL—Transport select user-defined objects (as specified on object control cards) from or into the TDF

Note: CA Telon internal objects such as user profiles and individual custom code members require object control cards to be transported.

zzzzzz

Columns 19-25. Specifies the processing option. For an export, use the processing option field to specify whether the system should perform auto-reference processing. For an import, use this field to specify either auto-merge processing or what to do if a specified object already exists on the TDF.

Valid values depend on the transport function.

n.n

Columns 66-68. Specifies the release number of the version of CA Telon used to transport out these objects. You may not transport-in a file with an older version of CA Telon, for example, you may not transport in a 5.1 file with r4.1.

Function	Valid Value	Meaning
Transport out	AUTOREF	Automatically transport out referenced items when primary items are specified (auto-reference processing).
	NOREF	(Default) Do not use auto-reference processing.
Transport in	IGNORE	Do not replace an existing object, but add the specified object if it does not exist on the TDF
	REPLACE	Replace an existing object, or add the specified object if it does not exist on the TDF.
	AUTOM*nn*	Use auto-merge processing with a maximum security level of nn.

u(40)

Columns 26-65. Specifies the ID of the user creating the transport file. This is an optional value that is valid on transport-out only. If supplied on the TRANSPORT control card, it appears in the header record of the transport file and appears on Transport status and summary reports.

Auto-reference processing

When you request auto-reference processing, the system automatically transports the global (Data Administration) objects referenced in the program's Data Group, as shown in this table:

Transported Item	Referenced In:	Time of Transport
IMSDBD TABLE or JOIN FILE CICSQUE CICSJRNL	Program Data Group	During program object transport*
IMSDBD TABLE or JOIN FILE CICSQUE	PSB or FILEGRP	During PSB or FILEGRP object transport

Transported Item	Referenced In:	Time of Transport
CICSJRN		
Physical databases or secondary indexes	IMSDBD	During IMSDBD object transport
TABLE	JOIN	During JOIN object transport

* If you specify auto-reference processing when transporting out a program, you can supply any of the object control cards that specify a program, for example PROGRAM and ONLINE.

Auto-merge processing

Auto-merge processing replaces (overlays) or merges global (Data Administration) objects in the TDF that are referenced in the program's Data Group on transport ins. You can specify auto-merge processing if the transport file was created by auto-reference processing.

Request auto-merge processing by specifying AUTOM*nn*, where *nn* specifies the maximum severity level. Valid *nn* values are:

Maximum Severity Level	Meaning
0	(Default.) No differences
2	Minor differences
4	Adding new DSCs
8	Minor inconsistencies
12	Structural differences
16	Major inconsistencies

Note: You can override the maximum severity level on individual object control cards.

When using auto-merge, CA Telon replaces the TDF object after comparing the transport file and the TDF object. Then CA Telon checks the return code for the compare:

- If the return code is less than or equal to the specified maximum severity level, Transport merges the TDF object with the transport file object. Null TDF data is changed to transport file values; non-null TDF values are not changed.
- If the return code for the compare is greater than the specified maximum severity level, Transport does not transport-in that object or any subsequent one.

Note: For more information about auto-reference and auto-merge processing, see the appendix "Managing Multiple TDFs."

User ID

Columns 26-65—The user ID is optional and is used only during transport-out processing. The value that you supply on the TRANPORT control card goes on the Transport Header record of the transport file. On import, it is ignored. This field is useful for understanding who created the transport file. It also appears on the Transport Status transport-in and Summary Reports.

Object Control Cards

The format of the object control cards is as follows:

```
----+----1----+----2----+----3----+----4----+----5----+----6-----
nnnnnnnn kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk ppppppps
```

The fields on the object control cards are described next.

nnnnnnnn

Columns 1-8—Identifies the kind of object to be moved. Valid values are:

- PANEL
- ONLINE
- BATCH
- DRIVER
- REPORT

- PROGRAM
- NONTERM
- STORED
- PSB
- FILEGRP
- USERPROF
- INSTALL
- PASSWORD
- MEMBER
- IMSDBD
- TABLE
- JOIN
- FILE
- CICSQUE
- CICSJRNL
- PSTORE

kkk...

Columns 10-49—Identifies the specific object to be moved. The length and contents of the field vary by object type.

You can supply one of these types of values:

- * (or *.* for TABLEs or JOINs)—Causes all key values for the identified object to be processed.
- Exact value—Specifies that the object identified by the exact value is processed.
- Partial key—Indicates a partial key followed by a wildcard character (*) that causes processing of all key values that begin with the same partial key. For example, ABCD* causes all key values beginning with ABCD to be processed. In a partial key value, the asterisk represents one or more characters.
- Wildcard pattern—Specifies a combination of exact value and one or more wildcards, represented by an asterisk (*), that causes processing of all key values that match the pattern. For example, TP**LS causes processing of all key values beginning with TP followed by any two characters and ending with LS. In a wildcard pattern, each asterisk represents one and only one character.

ppppppp

Columns 51-57. (Transport import only.) Optional override processing option that directs Transport processing, as presented in the following table:

Value	Processing that Results
spaces	Use the processing option on the TRANSPORT control card
IGNORE	If the object does not exist on the TDF, add it. Otherwise, ignore it.
REPLACE	Replace the TDF object with the transport file version.
BYPASS	Do not process the object.
* CHECK * MERGE * REFRESH	See Data Administration Objects.

Note: * These values are valid only for data administration objects IMSDBD, TABLE, JOIN, FILE, CICSQUE, CICSJRNL and IDMSDESC.

SS

Columns 58-59. (Import only.) The maximum acceptable return code for a Transport merge request. Valid only for Data Administration objects IMSDBD, TABLE, JOIN, FILE, CICSQUE, and CICSJRNL.

Object control card example

The next example shows valid object control cards. *Ppppppp* indicates a type of control card that can have a processing option override. *Ss* indicates a type of control card that can have a maximum severity level.

```
----+----1----+----2----+----3----+----4----+----5----+----6

PANEL    hhiiii                                  ppppppp
ONLINE   hhiiii                                  ppppppp
BATCH    hhiiii                                  ppppppp
DRIVER   hhiiii                                  ppppppp
REPORT   hhiiii                                  ppppppp
PROGRAM  hhiiii                                  ppppppp
NONTERM  hhiiii                                  ppppppp
STORED   hhiiii                                  ppppppp
IMSDBD   key(8)                                  pppppppss
TABLE    qual(8) table(18)                       pppppppss
TABLE    qual.table                              pppppppss
JOIN     qual(8) table(18)                       pppppppss
JOIN     qual.table                              pppppppss
FILE     key(8)                                  pppppppss
CICSQUE  key(8)                                  pppppppss
CICSJRNL key(8)                                  pppppppss
PSB      key(8)                                  ppppppp
FILEGRP  key(8)                                  ppppppp
PSTORE   key(8)                                  ppppppp
USERPROF key(8)                                  ppppppp
INSTALL  password(5)                             ppppppp
PASSWORD password(5)
MEMBER   hhiiii.xx(member(8))                    ppppppp
```

Each object control card is described next.

PANEL

Function

Transports panel definitions.

Key

Contains the panel header and ID. The key may be up to six bytes long.

Values

Valid override processing option values during the import are as follows:

spaces

Use the default processing option from the TRANPORT control card.

IGNORE

Ignore the contents of the transport file if the object already exists in the TDF.

REPLACE

Replace the TDF object.

BYPASS

Bypass the panel (or range of panels or subset of panels) on the transport file whether or not the object exists in the TDF-the equivalent of unconditional ignore processing.

ONLINE

Function

Transports online program definitions (*object-key*.PD and *object-key*.PI).

Key

Allows the same header and ID format as the PANEL control card. The key may be up to six bytes long.

Values

Allows the same override processing options as the PANEL control card.

BATCH

Function

Transports batch program definitions (*batch-program header+ID*.BD).

Key

Allows the same header and ID format as the PANEL control card. The key may be up to six bytes long.

Values

Allows the same override processing options as the PANEL control card.

DRIVER

Function

Transports IMS driver program definitions (*driver-key*.DR).

Key

Allows the same header and ID format as the PANEL control card. The key may be up to six bytes long.

Values

Allows the same override processing options as the PANEL control card.

REPORT

Function

Transports report program definitions (*report-key*.RD).

Key

Allows the same header and ID format as the PANEL control card. The key may be up to six bytes long.

Values

Allows the same override processing options as the PANEL control card.

NONTERM

Function

Transports CICS nonterminal program definitions (*nonterm-key*.ND).

Key

Allows the same header and ID format as the PANEL control card. The key may be up to six bytes long.

Values

Allows the same override processing options as the PANEL control card.

STORED

Function

Transports Stored Procedure program definitions (stored-key.SP)

Key

Allows the same header and ID format as the PANEL control card. The key may be up to six bytes long.

Values

Allows the same override processing options as the PANEL control card.

PROGRAM

Function

Transports any or all versions of a program definition (for example, online, batch, driver, nonterminal, and report). The panel (if any) is transported out once for each version of the program.

Key

Allows the same header and ID format as the PANEL control card. The key may be up to six bytes long.

Values

Allows the same override processing options as the PANEL control card.

Data Administration Objects

Data Administration Objects include the following object types:

- IMSDBD
- TABLE
- JOIN
- FILE (VSAM or sequential)
- CICSQUE
- CICSJRNL

Each Data Administration Object control card allows the same override processing options as the PANEL control card. It also allows these values during import processing to compare the Data Administration Object on the transport file against the corresponding Data Administration Object on the TDF and produce a report identifying any differences on the Change Impact/Change Analysis Report. The result of import processing is different for each option. The options, including their function, key, and values are described next.

- CHECK—The contents of the Data Administration Object from the transport file are not used to update the TDF. Use the CHECK option to assess the result of performing import processing with the Data Administration Object.
- MERGE—The contents of the Data Administration Object from the transport file are merged with the data already on the TDF if the severity level returned by the compare process is less than or equal to the specified maximum severity level. This means that null TDF data is changed to the transport file values, while non-null TDF values are changed.

  If the severity level returned from the compare process is greater than the maximum allowed, the Transport job step sets a condition code of 4. This is an informational code meaning that you should examine the Transport Status and the Data Administration Reports.

  When you request auto-merge, merge processing is the default if a Data Administration Object exists in the TDF. If not, the Data Administration Object is added to the TDF.
- REFRESH—The contents of the Data Administration Object from the transport file are used to replace the TDF if the severity level returned by the compare process is less than or equal to the specified maximum severity level.

  If the severity level returned from the compare process is greater than the maximum allowed, the Transport job step sets a condition code of 4. This is an informational code meaning that you should examine the Transport Status and the Data Administration Reports.

Note: CHECK, MERGE, and REFRESH are valid only for Data Administration objects (IMSDBD, TABLE, JOIN, FILE, CICSQUE, and CICSJRNL). When you specify one of these options, you should also specify a maximum severity level.

For more information about CHECK, MERGE, and REFRESH,.see the appendix "Managing Multiple TDFs."

IMSDBD

Function

Transports IMS DBDs.

Key

Specifies the name of the IMSDBD as entered. The key can be one to eight bytes long.

Values

Specifies that the IMSDBD control card allows the Data Administration Object override processing options, which include the same override processing options as the PANEL control card, plus CHECK, MERGE, and REFRESH.

TABLE

Function

Transports DB2 tables.

Key

Lets you supply the key in one of these ways:

- Enter a one- to eight-byte qualifier in columns 10-17 followed by a one- to 18-byte table name beginning in column 18
- The concatenated DB2 table name *qualifier.tablename*

Shown next are examples of valid key specifications for a table or join:

```
       0       1       1               3
Column 1........0.......8................5.
       TABLE   GUS.DB2
       TABLE   GUT     DB2
       JOIN    LONGNAME.TABLENAMEEIGHTTEEN
       JOIN    LANGNAMETABLENAMEEIGHTTEEN
       TABLE   GUS.EIGHTTEENTABLENAME
       JOIN    GUT     EIGHTTEENTABLENAME
```

Values

Specifies that the Table control card allows the Data Administration Object override processing options, which include the same override processing options as the PANEL control card, plus CHECK, MERGE, and REFRESH.

JOIN

Function

Transports joined tables.

Key

Lets you supply the key in the same way as on the TABLE control card.

Values

Specifies that the Join control card allows the Data Administration Object override processing options, which include the same override processing options as the PANEL control card, plus CHECK, MERGE, and REFRESH.

FILE

Function

Transports VSAM or sequential files.

Key

Has key from one to eight bytes long.

Values

Specifies that the file control card allows the Data Administration Object override processing options, which include the same override processing options as the PANEL control card, plus CHECK, MERGE, and REFRESH.

CICSQUE

Function

Transports CICS queues.

Key

Has key from one to eight bytes long.

Values

Specifies that the CICSQUE control card allows the Data Administration Object override processing options, which include the same override processing options as the PANEL control card, plus CHECK, MERGE, and REFRESH.

CICSJRNL

Function

Transports CICS journals.

Key

Has key from one to eight bytes long.

Values

Specifies that the CICS journal control card allows the Data Administration Object override processing options, which include the same override processing options as the PANEL control card, plus CHECK, MERGE, and REFRESH.

PSB

Function

Transports IMS PSB definitions.

Key

Has key from one to eight bytes long.

Values

Allows the same override processing options as the PANEL control card.

FILEGRP

Function

Transports file group definitions.

Key

Has key from one to eight bytes long.

Values

Allows the same override processing options as the PANEL control card.

PSTORE

Function

Transports presentation stores (made using the prototyping facility).

Key

Has key from one to eight bytes long.

Values

Allows the same override processing options as the PANEL control card.

INSTALL

Function

Transports the TDF installation defaults and security information.

Key

Contains only one INSTALL object. Instead of an object key, code the password of the TDF from or into which you are transporting.

Values

Allows the same override processing options as the PANEL control card.

USERPROF

Function

Transports CA Telon TDF user profile information.

Key

Has key from one to eight bytes long.

On a TSO TDF, the key for user profiles is a TDF user's TSO logon ID. On a CICS TDF, the key for user profiles is either the CICS operator ID assigned to a user or the CICS terminal ID.

Values

Allows the same override processing options as the PANEL control card.The USERPROF control card allows the same override processing options as the PANEL control card. During import processing, the override processing option can also contain a value of CHANGE. To transport a user profile from a mainframe TDF (where the key is a user ID) to a single-user PWS TDF, use the CHANGE processing override so that the user profile will be imported onto the PWS TDF under the TELONPC key.

Note: To transport user profiles successfully, you must supply the password for the TDF you are transporting against. You can do this by using the INSTALL control card or the PASSWORD control card.

PASSWORD

Function

Supplies the TDF password for use in the transport of user profiles.

Key

Specifies the one- to eight-byte password of the TDF against which you are transporting.

To transport user profiles successfully, you must supply this control card or the INSTALL control card. You can place this control anywhere in the Transport control file.

MEMBER

Function

Transports individual custom code members associated with a program definition.

Key

The key is:

xxxxxx.*yy*(*zzzzzzzz*)

where:

- *xxxxxx* is the one- to six-character concatenation of header and ID
- *yy* is the program type:
  - BD—Batch Program Definition
  - RD—Report Program Definition
  - ND—CICS Nonterminal Definition
  - DR—IMS Driver Program Definition
  - SD—Online Program Definition
  - SP—Stored Procedure Program Definition
- *zzzzzzzz* is the name of the custom code member

Example: TRMENU.SD(OINIT1)

For transport-out, you must supply exact key values; range and wild card keys are ignored. For transport-in processing, you can supply any type of key.

Values

Allows the same override processing options as the PANEL control card.

END Control Card

The format of the END control card is shown next.

```
----+----1----+----2----+----3----+----4----+----5----+----6-----
END
```

The fields for the END control card are described following.

END

Columns 1-3. Terminates the Transport control file.

When the system encounters the END control card, it terminates processing of the Transport control file and invokes the actual Transport process.

The system ignores any control cards following the END card and it ignores any values supplied on the END control card.

System Processing Using Control Cards

Transport control cards give you a lot of flexibility for transporting objects into and out of a TDF. This section describes the control card options available to you.

Process the Entire Transport File

To import a vendor-supplied transport file, specify the entire transport file to be loaded into the TDF. This is the default.

If an object does not exist on the TDF, the object is added. If it already exists on the TDF, two options are available:

- Replace the TDF contents with the object on the transport file
- Ignore the contents of the transport file (this is the default)

Use caution when you replace an object. If someone has changed or added information to the object on the target TDF, that information is lost.

Process Selective Objects

To process selected objects, supply one or more control cards for each object type. These control cards specify the actions to be performed on a specific object, group of objects, or subset of objects.

To perform the selective replacement of objects on the TDF, choose one of these methods:

- Perform a selective import, specifying the names (either exact names, ranges, wild cards, or a special key of ALL, or an asterisk) of the object(s) to be replaced or ignored
- Perform a non-select import and specify the names of the objects to be bypassed on the transport file

With Transport files, you have several additional options described in the following paragraphs.

Processing all objects

To move the entire contents of one TDF to another TDF, perform a non-selective transport-out from the source TDF and a non-selective import to the target TDF. On transport-out, the system places the entire user-defined contents of the TDF on the transport file.

Processing internal objects

Always transport CA Telon internal objects (for example, install or user profile) using the selective Transport features. You must supply the password of the source or target TDF before performing a transport-in or transport-out function for internal objects because of the sensitivity of the information and the importance of importing it correctly.

Multiple-TDF object processing

If users are developing a system on multiple TDFs (for example, in a PWS configuration), an individual user may need to Transport a specific program or group of programs to a master TDF for program generation. If the program contains data access requests, the user must transport the DBDs, SQL tables, or files so that CA Telon generates the program correctly. Similarly, for PSBs and file groups, referenced DBDs, SQL tables, and files must be transported so that the PSB or file group may be used.

Selective Transport-Out and Transport-In

To selectively transport out, specify the names of the programs, PSBs, or file groups and their associated DBDs, SQL tables, and files. If a DBD is for a logical database, you must also transport its physical databases. Similarly, you must transport additional information for the component SQL tables when referencing a joined SQL table.

For transport-in, there are additional considerations. If DBD or SQL tables are replaced on the target TDF, it is possible that any other programs which access these entities will no longer generate correctly. If DBDs, SQL tables, or files are *not* replaced on the target TDF, a transported program might not generate correctly.

Auto-reference processing

When performing selective transport-out operations, you can specify auto-reference processing to automatically transport a DBD, SQL table, CICS queue, CICS journal, or file referenced by a program, PSB, or file group. If the DBD is a logical database, the system automatically transports its physical databases. Similarly, the Transport system automatically transports the base tables for joined SQL tables.

To use the auto-reference option, specify the names of the programs, PSBs, or file groups to be transported.

Auto-merge processing

On import, you can specify auto-merge processing on a selective or non-selective transport. The system adds DBDs, SQL tables, CICS queues, CICS journals, or files to the target TDF if they do not already exist. If they do exist, they are merged with the target TDF's definition.

Before merging a DBD, SQL table, CICSQUE, CICSJRNL, or file object, the system compares the two objects to see if they are compatible and produces a change analysis report showing the results of object comparison. If the objects are compatible, depending on the severity level from the compare, they are merged. (When you specify auto-merge processing, you supply a maximum severity level for the compatibility between the two objects before the merge takes place.)

The system must process DBDs and SQL tables in a certain sequence for the merge to work properly; that is, the system must merge the physical DBDs or base tables before the logical DBDs or joined tables. When you request transport-out with auto-reference processing, the system creates the transport file in the correct sequence.

If auto-merge processing fails due to incompatibilities on a physical DBD or base table, the system does not perform a merge on any logical DBDs or joined tables on the transport file. Similarly, if processing fails for a logical DBD or joined table, then the system does not perform import processing for any programs, PSBs, or file groups on the transport file. If merge processing is successful, the system adds any programs, PSBs, and File Groups on the transport file to the target TDF or replaces them if they already exist on the TDF.

Advanced merge processing

You can perform advanced merge processing by specifying maximum severity levels on object control cards. The level may be higher or lower per object (or group or subset of objects) rather than for the entire auto-merge operation.

When not performing auto-merge processing, you can also use object control cards to check a DBD, SQL table, CICSQUE, CICSJRNL, or file object (or group or subset of objects) against the contents of the target TDF and obtain a change impact report of the differences for each DBD, SQL table, or file.

You can also refresh (totally replace) a DBD, SQL table, CICSQUE, CICSJRNL, or file in a target TDF and still obtain a change impact report. The system conditionally performs the refresh process based on the maximum severity level supplied by the user.

Automatic system checks

On transport-in, Transport automatically checks to see if the object is in use on the target TDF (for example, if a user is editing it). If so, import of the object does not occur.

If you use a user-supplied security routine in the TDF, transport processing calls it before each object is transported out or in. If the security routine denies access to the object, the requested processing does not occur.

Note: During auto-merge processing, if an object is in use or the security routine denies access, it is treated as a merge failure. Transport reports its processing actions on a status report. When you check, merge, or refresh DBDs, SQL tables, CICSQUEs, CICSJRNLs, IDMSDESCs, or files, the system produces a separate Change Impact and Change Analysis report.

Chapter 4: Transport Status Report

The Transport Status Report (TPRPT) reports the actions taken by the Transport system and reports the detected errors.

Transport-in processing of an IMSDBD, SQL table or join, CICS queue, CICS journal, or a File object also produces a Data Administration Report, described in the appendix "Sample Data Administration Report" to reflect data administration change analysis and impact.

This chapter describes the Transport Status Report with the following topics:

- Status Report file contents
- Transport processing statuses

Status Report File Contents

The Transport Status Report is produced when you perform a Transport import. It consists of a summary page and a detail processing section. The detail section lists by object, the processing performed by Transport. The summary page lists the files used during the Transport process. It also lists, by object type, a summary of the number of objects that the system processed and objects that the system did not process.

A sample Transport Status Report follows, including a description of each section.

```
DATE 04/01/05          TELON Transport                Page 1
TIME 11:15:47          FUNCTION: Transport-in
                       BY: Userid

OBJECT   NAME                    STATUS                RECORDS

CICSQUE  JAPQ              Merge Performed            12
                           User Supplied Merge Level  12
                           Returned Merge Level       00
CICSJRNL JAPJRNL           Merge Performed            11
                           User Supplied Merge Level  12
                           Returned Merge Level       12
NONTERM  GRND01.ND         Replaced                   119
PANEL    TRMENU            Added                      100
ONLINE   TRMENU.SD         Replaced                           1,000
BATCH    TRXX30.BD         Bypassed due to Collision          1,000
DRIVER   TRDRVR.DR         Bypassed on Transport file         500
REPORT   TRRPTI.RD         Not Found
PROGRAM  TRDISP.SD         Security Violation
                           ** USER  NOT AUTHORIZED
IMSDBD   TRGEMPL           Compare Performed                  100
                           Maximum Severity Level:    00
                           Reported Severity Level:   00
TABLE    TELON   TRGEMPL2  Bypassed - Object Incomplete       120
TABLE    TELON   TRGEMPL1  Merge Performed                    500
                           Maximum Severity Level:    04
                           Reported Severity Level:   00
JOIN     TLNJOIN TRGEMPL1  Merge Requested not Performed     50
                           Maximum Severity Level:    00
                           Reported Severity Level:   04
FILE     TRGEMPL1          Refresh Performed                  100
                           Maximum Severity Level:    16
                           Reported Severity Level:   08
PSB      TRPSB             Merge Failure, Bypassed            300
FILEGRP  TRFILEG           Bypassed in use by - KIRK1         350
PSTORE   TRADD             Bypassed via Control Card          200
USERPROF KIRK1             No Password Supplied               200
INSTALL                    Invalid Password                   200
```

Common Header

Both sections of the report use a common header that includes the processing dates and start time, the transport function (transport-in or transport-out), and the user ID. This user ID is obtained from the transport control file for a transport-out and the transport header record for a transport-in.

Detail Processing Section

The detail section of the report identifies the object type, name, Transport processing status, and number of subordinate records processed for the object.

Summary Page

The summary page contains two sections: File Names and Object Summary.

File Names Section

The File Names section lists the names of the files used and produced during the transport process. Those files are as follows:

Transport file

Input file for Transport import, output file for Transport export

Status report file

Destination of the status report

Control file

Governs the Transport processing

DA report file

Report on Data Administration processing

TDF files

TNTDF, TNTDD, TNTDX, TNTCCL

Object Summary Section

The Object Summary section lists the number of objects processed for that object type. This section lists the object types in the order processed. For each object type listed, the report shows the number of objects that the system processed and did not process.

```
DATE 04/01/05          TELON Transport              Page 2
TIME 11:15:47          FUNCTION: Transport_In
                       BY: Userid

                          SUMMARY PAGE

 Transport file Name: TELON.PROD.TRANPORT (TRANIN)
 Status Report Name:  TELON.PROD.TRANPORT (TRANRPT)
 Control File Name:   TELON.PROD.TRANPORT (TRANCTRL)
 DA Report File Name: TELON.PROD.TRANPORT (TRANMSG)
 TNTDX:               VTELON.PROD.TNTDX
 TNTDD:               VTELON.PROD.TNTDD
 TNTDF:               VTELON.PROD.TNTDF
 TNTCCL:              VTELON.PROD.TNTCCL

 OBJECT           PROCESSED        NOT PROCESSED
 ______           _________        _____________

 CICSQUE                1
 CICSJRNL               1
 NONTERM                1
 PANEL                  1
 ONLINE                 1
 BATCH                                       1
 DRIVER                                      1
 REPORT                                      1
 PROGRAM                                     1
 IMSDBD                 1
 TABLE                  1
 JOIN                                        1
 FILE                   1
 PSB                                         1
 FILEGRP                                     1
 PSTORE                                      1
 INSTALL                                     1
 USERPROF                                    1
```

Transport Processing Statuses

The following table describes the values that Transport can assign to the processing status. Transport-in and Transport-out produce different status values.

The table lists the function (IN for transport in, OUT for transport out) that can issue the status value, whether the object was processed (YES/NO/BOTH), status values, and a description of the status.

Function	Processed	Status message/description
IN	YES	Added Object did not exist and was added to the TDF.
IN	YES	Replaced Object existed on the TDF and was replaced (Processing Option REPLACE or auto-merge).
IN	YES	Compare Performed Compare processing for an IMSDBD, TABLE, JOIN, CICSQUE, CICSJRNL, or FILE was requested and performed. This status message is followed by two additional message lines: ■ Maximum Severity Level: nn ■ Reported Severity Level: nn The first line identifies the user-supplied Maximum Severity Level. The second line identifies the severity level that the compare process reported. The system also produces a Data Administration Report for the object (see the appendix "Sample Data Administration Report").

Function	Processed	Status message/description
IN	YES	Merge Performed Compare processing for an IMSDBD, TABLE, JOIN, CICSQUE, CICSJRNL, or FILE was requested and performed. This status message is followed by two additional message lines: ■ Maximum Severity Level: nn ■ Reported Severity Level: nn The first line identifies the user-supplied Maximum Severity Level. The second line identifies the severity level that the compare process reported. The system also produces a Data Administration Report for the object (see the appendix "Sample Data Administration Report").
IN	YES	Refresh Performed Compare processing for an IMSDBD, TABLE, JOIN, CICSQUE, CICSJRNL, or FILE was requested and performed. This status message is followed by two additional message lines: ■ Maximum Severity Level: nn ■ Reported Severity Level: nn The first line identifies the user-supplied Maximum Severity Level. The second line identifies the severity level that the compare process reported. The system also produces a Data Administration Report for the object (see the appendix "Sample Data Administration Report").
IN	NO	Bypassed due to Collision Object already exists on the TDF (Processing Option IGNORE).
IN	NO	Bypassed on Transport file An object was found in the transport file for which you did not request transport processing.
IN	NO	Not Found Objects requested in control cards were not found on the transport file.

Function	Processed	Status message/description
IN	NO	Bypassed in use by - userid Another user is currently updating the object. The user ID identifies the user who is updating it. **Note:** For PSTORE objects, no user ID is available.
IN	NO	Bypassed via Control Card An object on the transport file with override processing option BYPASS.
IN	NO	DBDs Incompatible Compare or merge processing was requested for an IMSDBD and was not performed because the DBD's hierarchies were not compatible.
IN	NO	Compare not Performed Compare processing was requested for an IMSDBD, TABLE, JOIN, CICSQUE, CICSJRNL, or FILE and was not performed because the IMSDBD, TABLE, JOIN, CICSQUE, CICSJRNL, or FILE does not exist on the Target TDF.
IN	NO	Merge Requested not Performed Merge processing for a IMSDBD, TABLE, JOIN, CICSQUE, CICSJRNL, or FILE was requested but not performed because the reported severity level was greater than the user-supplied Maximum Severity Level. This status message is followed by two additional message lines: ■ Maximum Severity Level: nn ■ Reported Severity Level: nn The first line identifies the user-supplied Maximum Severity Level. The second line identifies the severity level that the compare process reported. The system also produces a Data Administration Report for the object (see the appendix "Sample Data Administration Report").

Function	Processed	Status message/description
IN	NO	Refresh Requested not Performed Refresh processing for a IMSDBD, TABLE, JOIN, CICSQUE, CICSJRNL, or FILE was requested but not performed because the reported severity level was greater than the user-supplied Maximum Severity Level. This status message is followed by two additional message lines: ■ Maximum Severity Level: nn ■ Reported Severity Level: nn The first line identifies the user-supplied Maximum Severity Level. The second line identifies the severity level that the compare process reported. The system also produces a Change Impact/Change Analysis Report for the object.
IN	NO	Merge Failure, Bypassed Auto-merge transport-in processing failed for one or more IMSDBDs, TABLEs, JOINs, CICSQUEs, CICSJRNLs, or FILEs. The system did not process the object identified. Failure can occur because of a merge processing failure, an object being in use, or a Security Violation.
IN	NO	Program does not exist You are attempting a transport-in of an object but the application program does not exist on the Target TDF. The system performed no processing for the object.
BOTH	NO	Security Violation The user-supplied security routine denied access to the object. The security routine may issue another message following the status message.
IN	NO	BYPASSED - Object Incomplete Attempt to transport in a data access object (SQL TABLE, JOIN, or IMSDBD) containing only the root object and none of the dependent objects information failed. Such a 'dummy' object may not be transported in as doing so would cause serious problems when accessed on the TDF.

Function	Processed	Status message/description
BOTH	NO	Invalid Password An invalid TDF password for an INSTALL object or on a PASSWORD control card was supplied. If it was supplied on an INSTALL control card, the system did not process the INSTALL object.
BOTH	NO	No Password Supplied An attempt was made to transport user profiles with no TDF password (or an invalid TDF password) supplied. User profiles are not transported.
OUT	NO	Not Found Transport objects using control cards not found on the source TDF.
OUT	YES	(No message) Object was successfully transported.

Chapter 5: Import/Export Procedures

The INSTALL PDS includes JCL and PROC members, and procedures used to perform common CA Telon tasks. Some of these are discussed and tested as part of the installation process described in the *Installation Guide*.

The #CUSTJCL process can be used to create a collection of JCL and PROC members in the *pdsqual*.PROCPDS files using the #CUSTCTL member as input.

This chapter describes how to use these members to:

- Export, generate, compile, and link TDF defined programs
- Export TDF program or panel to a source code library
- Import CA Telon source code into a TDF
- Import existing DBDs and PSBs into the TDF's Data Administration

Each of these operations requires you to set several parameters in the procedure before submitting the related job. Many of these parameters are set during the #CUSTJCL customization as described in the *Installation Guide*. You will have to change others each time you use the procedures.

Export, Generate, Compile, and Link

The JCL and PROCs created in *pdsqual*.PROCPDS can be used to export CA Telon definitions from the TDF creating CA Telon source code, to generate a program, and to compile and link load modules.

A JCL and PROC pair is generated for each combination of language, target environment, and DBMS you request in process (for example, CICS-PL/I-VSAM or IMS/DC-COBOL-DB2).

Many of the parameters used by these PROCs and JCL are changed during the configuration portion of the install process. For more information about this, see the Installation Guide.

The following jobs (along with their associated procedures) are a representative sample of the PROCs used to:

- Export a CA Telon TDF program
- Generate, compile, and link a completed COBOL or PL/I program from that exported CA Telon source

Target Env	Job Name	Procedure	Language
Batch	JNBXGCL	TLNBXGCL	COBOL
CICS	JNCXGLP	TLNCXGPL	PL/1
IMS	JNIXG2L	TLNIXG2L	COBOL II
IMS (w/DB2 support)	J2IXGCL	TL2IXGCL	COBOL

Parameters for PROCs

To export, generate, compile, and link a TDF program, make sure the following parameters in the appropriate procedure are set to the required values, then submit the correct job. Default values are in italics.

Note: An asterisk (*) preceding a parameter indicates that the parameter is set by #CUSTCTL to installation defaults as described in the *Installation Guide*.

Parameters

CICS

Parameter	Required Values
BMSMEM=member-name	The name of the CA Telon-generated BMS member for the program.
BMSSRC=lib-name	The name of the source library into which the BMS source name specified in the BMSMEM parameter will be written.
*CICSCPY=lib-name	The name of your CICS copy library that holds the layout for DFHEIB.
*CICSLIB=lib-name	The name of the CICS load library that holds members such as DFHEIC and DFHEIP.
*ECPPARM='parm1 {,parm2}...'	CICS Command translator (ECP) parameters to be added to those hard-coded into the ECP (CICS precompile) step by #CUSTCTL. Parm2 is used for CICS COBOL II.

DB2 Only

Parameter	Required Values
DBRMLIB=dbrm library name	The library that will contain the DBRM member.
DBRMMEM=dbrm-member-name	The DB2 DBRM source member name. You must supply a value for this parameter.

IMS/DC

Parameter	Required Values
*IMSRES=lib-name	The DL/I load library.
MFSMEM=member-name	The CA Telon-generated MFS source member to be created for this program.
MFSSRC=lib-name	The source library into which the MFS source member specified in the MFSMEM parameter will be written.

All Environments

Parameter	Required Values
CMP=C or N	Indicates whether the CA Telon source code is to be compressed. Values are: ■ C—Compress the source code; that is, put as many parameters as possible on a line ■ N—Write only one parameter on a line
*COBLIB=lib-name	The name of your COBOL II or COBOL for z/OS load library.
*CPARM='parm1-,parm2-...'	COBOL compile parameters to be added to parameters hard-coded for the COBOL compiler by #CUSTCTL.
CPYLIB1=lib-name CPYLIB2=lib-name CPYLIB3=lib-name	The names of three PDS copy libraries to be used during CA Telon resolve and compile steps. You do not have to code all three parameters.
CRDPARM=ADPCCARD parameter	This parameter should not be changed. It is used for CA Telon internal processing.
DBLSPC=Y/N	Double space. It is used to format a panel to print.
DEFTYPE=SD | | RD | | DR| | BD | ND | SP |	The type of TDF Definition being processed. Options are: ■ SD—Screen Definition—IMS/DC and CICS ■ RD—Report Definition (IMS/DC) ■ DR—Driver Definition (IMS/DC) ■ BD—Batch Definition (Batch Only) ■ ND—Non terminal Definition (CICS) ■ SP—Stored Procedure Definition

Parameter	Required Values
ENV=C |.B |I | T || S | R	The environment in which the program executes. Values are: ■ C—CICS ■ B—Batch ■ I—IMS/DC ■ T—TSO ■ S—CICS Client ■ R—Stored Procedure This parameter is used by Export to determine the environment for the program being exported.
FORMAT=B | N or FORMAT=M | N	The type of source mapping. Values are: **For CICS** ■ B—Create a BMS map ■ N—Do not create a BMS map **For IMS/DC-TSO** ■ M—Create an IMS MFS map source ■ N—Do not create a map source **Note:** To create a BMS or MFS map, you must specify a format value here and then supply the appropriate value for the OPTION parameter below (for example, FORMAT=B, OPTION=6 to generate a CICS BMS map).
LPARM='parm1-,parm2-...'	The linkage editor parameters to be added to the hard-coded list of linkage editor parameters. COBOL II, and COBOL for z/OS use different parameters.

Parameter	Required Values
OPTION=number	The process you want the procedure to perform. Values are: ■ 0—Generate a COBOL or PL/I source, compile, link and create BMS/MFS and PSB source ■ 1—Generate, compile, link, and create BMS/MFS source ■ 2—Generate, compile, link, and create BMS/MFS source ■ 3—Generate, compile, link, and create BMS/MFS source ■ 4—Generate, compile, link, and create BMS/MFS source ■ 5—Generate, compile, link, and create BMS/MFS source ■ 6—Generate and create BMS/MFS source ■ 7—Generate and create PSB source ■ 8—Generate For BATCH the only valid values are 2, 3, 4, 7, and 8. For Batch, the only valid values are 2, 3, 4, 7, and 8. **Note:** To generate a BMS or MFS map, you must also specify the correct FORMAT value. To create PSB source, you must also specify the correct PSB value.
*PANLIB=lib-name	CA Panvalet library where copybooks (to be resolved in programs) are stored.
*PANLOAD=lib-name	CA Panvalet load library
*PGMASM=asm-name	The 1-8 character name of the IBM assembler. CA Telon supports ASMA90.
*PGMCOB=cob-name	The 1-8 character name of the IBM COBOL compiler. Enter this parameter for either COBOL II or COBOL for z/OS (IGYCRCTL).
*PGMPL/I=pli-name	The name of the IBM PL/I compiler.
*PGMLNK=linker	The name of the IBM linkage editor.
*PLILIB=lib-name	The name of your PL/I load library.

Parameter	Required Values
*PPARM='parm[,parm2]...'	PL/I compile parameters to be added to parameters hard-coded in the PL/I compile step by #CUSTCTL.
PSB=D | N or I | N	This indicates whether you want PSB source code generated. Values are: **For CICS** ■ D—Create a DL/I PSB source ■ N—Do not create this source **For IMS/DC** ■ I—Create an IMS PSB source ■ N—Do not create this source **Note:** To create a PSB source, you must also specify the appropriate value for the OPTION parameter (for example, PSB=I, OPTION=7 to generate IMS PSB source code).
PSBMEM=member-name	The name of the CA Telon-generated PSB member for this program.
PSBSRC=lib-name	The name of the source library into which the PSB source specified in the PSBMEM parameter will be stored.
RESxxxx=various	A generator resolve option. There are thirteen resolve options that all start with RES.
*SCRUNIT=unit-name	The disk scratch unit.
*SRCLIB=libname	The data set name of the source library where CA Telon is to place the generated source code.
TDFMEM=hhnnnn	The name of the TDF member the procedure is to process according to the OPTION parameter. This is the Header-ID combination name used at your installation. hhnnnn (2-character header, 4-character ID) is the default.
TESTMAC=lib-name	CA Telon maclib intended to hold customized or test macros. It is concatenated in front of TLNMAC.
*TLNLOAD=lib-name	The name of the CA Telon load library.
*TLNMAC=lib-name	The name of the CA Telon macro library.
*USRLOAD=lib-name	The name of the load library where the generated load module is placed after it is link edited.

Parameter	Required Values
*USRSUBR=lib-name	The name of the load library containing user-written subroutines called by the generated program.
*VSQUAL=vsamqual	The high-level data set name qualifier for the TDF's VSAM data sets.

Export TDF Definition to a Library Source

The following jobs and procedures allow you to export a TDF program definition to create a CA Telon source code PDS, AllFusion Panvalet, or AllFusion Endevor Change Manager member:

To	Job	Procedure
A PDS	JUXDEF	TLNUXDEF
CA Panvalet	JUXPAN	TLNUXPAN
CA Software Change Manager Endevor	JUXDFE	TLNUXDFE

Note: By using Job JUXPAN that executes Procedure TLNUXPAN, you can put your CA Telon source into a CA Panvalet library. When the TDF is running under CICS, you cannot use the TDF Utilities menu (Option U) to export a program definition to a source code PDS or export to CA Panvalet from the TDF Utilities menu under either CICS or TSO.

Parameters for PROCs

To export a TDF program definition into CA Telon source code, make sure the following parameters in the procedure are set to their appropriate values; then submit the job.

An asterisk (*) precedes those parameters set by #CUSTCTL to installation defaults during the installation System Load. See the Installation Guide

All Target Environments

The following parameters are used in all target environments:

Parameter	Required values
CMP= C | N	Indicates whether the CA Telon source code is to be compressed or not. Values are:

Parameter	Required values
	■ C—Compress the CA Telon source code; that is, put as many parameters as possible on a line ■ N—Put only one parameter per line
DEFTYPE=SD | BD | PD | PI | RD | DR | ND | SP	The type of TDF Definition being processed into CA Telon source code. Values are: ■ SD—Screen Definition (default in IMS/DC and CICS) ■ BD—Batch Definition (Batch) ■ PD—Panel Definition ■ PI—Panel Image ■ RD—Report Definition (IMS/DC only) ■ DR—Driver Definition (IMS/DC only) ■ ND—Non terminal Definition (CICS only) ■ SP—Stored Procedure Definition
ENV=C|B|I|T|S|R	The target environment the program executes in. Values are: ■ C—CICS ■ B—BATCH ■ I—IMS/DC ■ T—TSO ■ S—CICS Client ■ R—Stored Procedure This parameter is used by Export to determine the desired environment for the program being exported.

Parameter	Required values
PSB=D | N or PSB=I | N	This indicates whether you want PSB source code generated. When executed, the Generator responds to the request and generates the actual source code. Options are: **CICS or BATCH only** ■ D—Generate CA Telon source code that requests the Generator to create DL/I PSB source code ■ N—Do not create this source code **IMS/DC or BATCH only** ■ I—Generate CA Telon source code that will request the Generator to create IMS PSB source code ■ N—Do not create this source code
TDFMEM=hhnnnn	The name of the TDF member that the procedure is to export. This is the Header-ID combination name used at your installation. hhnnnn is the default (that is, 2-byte header and 4-byte ID).
*TLNLOAD=libname	The name of the CA Telon LOAD library.
*VSDISP=vsdisp	The disposition of the TDF's VSAM data sets. Set this to one of the following: ■ VSDISP=OLD for TDF running under CICS. In this case, you must deallocate your CICS TDF VSAM files from CICS before exporting. ■ VSDISP=SHR for TDF running under TSO.
*VSQUAL=vsamqual	The high-level data set name for the TDF's VSAM data sets.

Import CA Telon Source Code into TDF

The following jobs and procedures allow you to import a CA Telon source code member into the TDF:

From	Job	Procedure
A PDS	JUMDEF	TLNUMDEF
CA Panvalet	JUMPAN	TLNUMPAN
CA Librarian	JUMLIBR	TLNUMLBR

From	Job	Procedure
Change Manager	JUMDFE	TLNUMDFE

Note: When the TDF is running under CICS, you must deallocate the TDF VSAM data sets to run this import function. Import sets the data sets to DISP=OLD.

We recommend that CA Telon source import procedures be utilized only after reading the appendix "Importing Data Inheritance" in the *Programming Concepts Guide* for specifics on import processing.

To translate CA Telon source code into its equivalent TDF design components, make sure the following parameters in the procedures are set to their appropriate values. Then submit the job.

An asterisk (*) precedes names set by #CUSTCTL to installation defaults during the installation System Load. For more information, see the Installation Guide.

All Environments

The following parameters are used for all environments:

Parameter	Required Values
*CTLLIB=lib-name	The data set name of the library containing control cards required to execute TDF utilities.
MAXSEVR=##	The two-digit indicator of the highest acceptable comparison severity. See the "Importing Data Inheritance" appendix in the *Programming Concepts Guide* for additional information.
*PANLIB=lib-name (CA Panvalet ONLY)	The data set name of your CA Panvalet source library.

Parameter	Required Values
RPTFMT=*L* or S	Indicator defining report format. If you specify a RUNTYPE of M or C and a RPTFMT of L, the differences between the imported source data and the TDF Data Administration data is reported. Values are: ■ L (*Default*)- Long Report. Prints differences. ■ S- Short Report. Does not print differences; prints a summary only.
RUNTYPE=I, M, or C	Indicator defining the scope of import processing. For more information, see the appendix "Importing Data Inheritance" in the *Programming Concepts Guide*.
*SRCLIB=lib-name	The data set name of the library holding CA Telon source code to be imported by this procedure.
SRCMEM=member-name	The data set name of the member within SRCLIB holding CA Telon source code to be imported.
*TESTMAC=lib-name	CA Telon maclib intended to hold customized or test macros. It is concatenated in front of TLNMAC.
*TLNLOAD=lib-name	The data set name of the CA Telon LOAD library.
*TLNMAC=lib-name	The data set name of the CA Telon MACLIB library.
*VSDISP=vsdisp	The disposition of the TDF's VSAM data sets. Set this to one of the following: ■ VSDISP=OLD for TDF running under CICS If the CICS TDF is active, you must deallocate your CICS TDF VSAM files from CICS before exporting. ■ VSDISP=SHR for TDF running under TSO
*VSQUAL=vsamqual	The high-level qualifier for the TDF VSAM data sets into which you want to import.

Import DBDs and PSBs

The following jobs and procedures allow you to import existing DBD and PSB source data into the TDF's data administration files. This makes these DBDs and PSBs available to TDF application development.

To Import from a	Job	Procedre
DBD	JUMDBD	TLNUMDBD
PSB	JUMPSB	TLNUMPSB

Note: When the TDF is running under CICS, you must deallocate the TDF VSAM data sets to run this import function. Import sets the data sets to DISP=OLD.

We recommend that CA Telon source import procedures be utilized only after reading the appendix "Importing Data Inheritance" in the *Programming Concepts Guide* for specifics on import processing.

You must have DBD and PSB source code for this operation. If your DBDs and PSBs are generated from your data dictionary without capturing source code, you have one of two options:

- Alter the existing JCL to capture that source code
- Merge the PROC steps that create the DBD and PSB source into the beginning of this TLNUMDBD procedure

An asterisk (*) precedes those parameters set by #CUSTCTL to installation defaults during the installation System Load. For more information, see the Installation Guide.

Parameters

IMS/DC-TSO only

Parameter	Required Values
*IMSMAC=lib-name	The data set name of the IBM-supplied macro library containing DBD macros

All Environments

These parameters are used in all environments:

Parameter	Required Values
CMP=C | N	Indicates whether the CA Telon source code is to be

Parameter	Required Values
	compressed. Values are: C—Compress the source code; that is, put as many parameters as possible on a line N—Put only one parameter on a line
*COBLIB=lib-name	The data set name of your COBOL II, or COBOL for z/OS load library.
*CPARM='parm1-,parm2-...'	COBOL compile parameters to be added to parameters hard-coded for the COBOL compiler.
CPYLIB1=lib-name CPYLIB2=lib-name CPYLIB3=lib-name	The data set names of three PDS copy libraries to be used during CA Telon resolve and compile steps. You do not have to code all three parameters.
CRDPARM=ADPCCARD parameter	This parameter should not be changed. It is used for CA Telon internal processing.
DBLSPC=Y/N	Double space. Used for a panel print.
DEFTYPE=SD | RD | DR | BD | ND | SP	The type of TDF Definition being processed. Options are: ■ SD—Screen Definition—IMS/DC and CICS ■ RD—Report Definition (IMS/DC) ■ DR—Driver Definition (IMS/DC) ■ BD—Batch Definition (Batch Only) ■ ND—Non terminal Definition (CICS) ■ SP—Stored Procedure Definition
ENV=C|B|I|T|S|R	The environment in which the program executes. Values are: ■ C—CICS ■ B—Batch ■ I—IMS/DC ■ T—TSO ■ S—CICS Client ■ R—Stored Procedure This parameter is used by Export to determine the desired environment for the program being exported
FORMAT=B|N or FORMAT=M|N	The type of source mapping. Values are: **For CICS** ■ B—Create a BMS map

Parameter	Required Values
	■ N—Do not create a BMS map **For IMS/DC-TSO** ■ M—Create an IMS MFS map source ■ N—Do not create a map source **Note:** To create a BMS or MFS map, you must specify a format value here and then supply the appropriate value for the OPTION parameter below (for example, FORMAT=B, OPTION=6 to generate a CICS BMS map).
LPARM='parm1-,parm2-...'	Linkage editor parameters to be added to the hard-coded list of linkage editor parameters. COBOL II, and COBOL for z/OS use different parameters.
OPTION=number	The process you want the procedure to perform. Values are: ■ 0—Generate a COBOL or PL/I source, compile, link and create BMS/MFS and PSB source ■ 1—Generate, compile, link, and create BMS/MFS source ■ 2—Generate, compile, link, and create BMS/MFS source ■ 3—Generate, compile, link, and create BMS/MFS source ■ 4—Generate, compile, link, and create BMS/MFS source ■ 5—Generate, compile, link, and create BMS/MFS source ■ 6—Generate and create BMS/MFS source ■ 7—Generate and create PSB source ■ 8—Generate For BATCH the only valid values are 2, 3, 4, 7, and 8. **Note:** To create a BMS or MFS map, you must also specify the correct FORMAT value. To create PSB source, you must also specify the correct PSB value.
*PANLIB=lib-name	CA Panvalet library where copybooks (to be resolved in programs) are stored.
*PANLOAD=lib-name	CA Panvalet load library
*PGMASM=asm-name	The 1-8 character name of the IBM assembler. CA Telon supports ASMA90.

Parameter	Required Values
*PGMCOB=cob-name	The 1-8 character name of the IBM COBOL compiler. Enter this parameter for either COBOL II or COBOL for z/OS (IGYCRCTL).
*PGMPL/I=pli-name	The name of the IBM PL/I compiler.
*PGMLNK=linker	The name of the IBM linkage editor.
*PLILIB=lib-name	The name of your PL/I load library.
*PPARM='parm[,parm2]...'	PL/I compile parameters to be added to parameters hard-coded in the PL/I compile step.
PSB=D|N or I|N	This indicates whether you want PSB source code generated. Values are: **For CICS** ■ D—Create a DL/I PSB source ■ N—Do not create this source **For IMS/DC** ■ I—Create an IMS PSB source ■ N—Do not create this source **Note:** To create a PSB source, you must also specify the appropriate value for the OPTION parameter (for example, PSB=I, OPTION=7 to generate IMS PSB source code).
PSBMEM=member-name	The name of the CA Telon-generated PSB member for this program.
PSBSRC=lib-name	The name of the source library into which the PSB source specified in the PSBMEM parameter will be stored.
RESxxxx=various	A generator resolve option. There are twenty resolve options that all start with RES.
*SCRUNIT=unit-name	The disk scratch unit.
*SRCLIB=libname	The data set name of the source library where CA Telon is to place the created source code.
TDFMEM=hhnnnn	The name of the TDF member the procedure is to process according to the OPTION parameter. This is the header-ID combination name used at your installation. hhnnnn is the default (that is, 2-byte header and 4-byte ID).
TESTMAC=lib-name	CA Telon maclib intended to hold customized or test

Parameter	Required Values
	macros. It is concatenated in front of TLNMAC.
*TLNLOAD=lib-name	The name of the CA Telon LOAD library.
*TLNMAC=lib-name	The name of the CA Telon MACLIB library.
*USRLOAD=lib-name	The name of the load library the newly generated load module is moved into after it is link edited.
*USRSUBR=lib-name	The name of the load library holding user-written subroutines called by the generated program.
*VSQUAL=vsamqual	The high-level qualifier for the TDF's VSAM data sets.

Additional PSB parameter

There is one more PSB parameter (in TLNUMPSB) you can set: CTLPARM. This parameter indicates where the procedure finds the PCBNAME within the PSB source being imported.

PCBNAME is moved into the REQUEST column in your data group in the TDF during import. You can also choose to manually enter this information, or not enter it at all.

CTLPARM does this by identifying:

- The record columns that will contain a unique literal (like, "*PCBNAME=")
- The columns where that literal is found

The value of CTLPARM is broken down into bytes.

Byte	Meaning
1 and 2	Column where the PCB name value begins.
3 and 4	Length of the PCB name value.
5	The number of unique search strings within the PSB source record. The record must match the following strings in order for this record to hold a valid PCB name.

The following groups of bytes are repeated once for each search string being identified.

Byte	Meaning
2 bytes	Starting column of search string
2 bytes	Length of search string

Byte	Meaning
n bytes	Value of search string

Example 1

To find a 12 character long PCB name starting in column 10 on a record starting with *PCBNAME= starting in column one, code:

```
CTLPARM='101210109*PCBNAME='
```

This is broken down as:

Code	Description
10	PCB name starts in column 10
12	Twelve characters long
1	Only one unique string
01	That string starts in column one
09	Nine bytes long
*PCBNAME=	The value of the string is *PCBNAME.

Example 2

A more complex example has blanks in the first two columns of a record, the string PCBGEN*PCBNAME= in columns 10 through 24 followed by an eight-character name. It is coded:

```
CTLPARM='250820102xx1014PCBGEN*PCBNAME='This is broken down as:
```

This is broken down as:

Code	Description
25	PCB name starts in column 25
08	Eight characters long PCB name
2	Record must have two unique strings
01	First record string starts in column one
02	First string is two bytes long
xx	Value of the string is two spaces
10	Second string starts in column 10
14	Second string is 14 bytes long

Code	Description
PCBNAME=	Value of the second string

Chapter 6: Automated Documentation

Automated Documentation allows you to generate reports that describe the contents of both the CA Telon Design Facility (TDF) VSAM files and the mainframe PDSes that are also associated with your CA Telon development system.

This chapter consists of the following sections:

- Facility Overview—Describes the CA Telon Automated Documentation Facility and provides a diagram showing the various parts of the facility.
- Extract Files—Describes how you can specify the contents of these files, and how these files are created and used.
- Standard Reports—Describes the jobs used to create the standard reports and details of how to specify the control cards to create the standard reports. the standard reports are described in detail, and sample reports are provided.
- Global Cross-Reference Reports—Shows how to generate and use these reports. The control cards and job parameters are described. The Global Cross-Reference reports are described in detail and sample reports are provided.
- Customized Reports—Gives a detailed description of the layout of the extract files so you can design additional reports as needed.

Facility Overview

To help maintain and control an application, you can generate documentation that organizes and summarizes such system characteristics as:

- Where and how to access tables, segments, and files
- What databases to access, and how to access them
- Where to use Custom Code
- What program definitions exist for an application
- What components of an application that's under development have been created
- What key stages of the application have been completed

The CA Telon Automated Documentation Facility enables developers and administrators to produce a variety of cross-reference style reports on the entities sorted in the CA Telon Design Facility application repository. This information is always current because the reports use the actual data that generated the application programs.

Terminology

This section provides definitions of common terminology in the CA Telon Automated Documentation Facility.

Definition of Entity

The term *entity* used throughout this chapter refers to a program, in whole or in part which has been designed under the TDF.

The Entity Name consists of three distinct parts:

- HEADER—One to five characters
- ID—One to five characters
- TYPE—Two characters

The length of the HEADER and ID is determined when the TDF is installed, but the total length of the HEADER and ID cannot exceed six characters.

A complete CA Telon program consists of:

- One or more distinct *entities* all having the same HEADER and ID
- One and only one entity of TYPE SD, DR, RD, BD, SP, or ND
- One entity each of TYPE PD and PI.

Note: The DR and SP have *no* PI or PD. The SD must have *both* a PI and a PD, required for SD and RD, optional for BD and ND, prohibited for DR and SP.

Autodoc Components

The CA Telon Automated Documentation Facility consists of the following components:

- Autodoc extract files
- Standard autodoc reports
- Global Cross-Reference Reports
- Customized Reports

The CA Telon Automated Documentation Facility components are illustrated and discussed next.

Autodoc Extract Files

The Autodoc Facility creates three extract files. Two of these files are used by Autodoc in the creation of the standard reports. The third file provides additional information regarding CA Telon TDF entities. All three files can be used to create additional, customized reports at your installation.

Note: For more information about how you can specify the contents of these files and how these files are created, see Extract Files.

Standard Autodoc Reports

The Automated Documentation Facility in CA Telon generates the following types of standard reports:

- Custom Code reports
- File and Database reports
- SQL Table reports
- Global Cross-Reference Reports
- Customized reports

Each type of standard report is described next.

Custom code Reports

- Custom Code Summary by Entity—Provides a report of all CA Telon TDF entities that reference Custom Code members
- Custom Code Cross-Reference—Shows the same information as the Custom Code Summary by Entity Report, but provides it in order by Custom Code member name so that you can see easily all places in which a Custom Code member is referenced

File and Database Reports

- Data Access Summary—Shows where a database or file name is referenced by a CA Telon TDF entity. This allows you to see, for each database or file defined to CA Telon, which entities reference it.
- Data Access Detail—Lists information about each type of file access for the databases or files referenced by CA Telon TDF entities. This allows you to see, for each database or file defined to CA Telon, which entities access the file and how they access it.
- Data Access Detail by Entity—Shows the same information as the Data Access Detail Report, but provides it in order by CA Telon TDF Entity. This report will allow you to review, by entity, what files and/or databases are accessed and how.

SQL table Reports

- SQL Table Name Cross-Reference—Provides a list that cross-references CA Telon Names used to identify SQL objects and the fully-qualified SQL Table and Join names. This list is ordered by SQL name.
- CA Telon Name Cross-Reference—Shows the same information as the SQL Table Name Cross-Reference, but it is ordered by the CA Telon name.

For more information, see Standard Reports.

Global Cross-Reference Reports

CA Telon's unique cross-reference reporting ability allows you to combine a scan of the CA Telon TDF with a scan of several separate PDSes and determine everywhere a COBOL or PL/I data element or any given character string is referenced.

For more information, see Global Cross-Reference Reports. For a detailed description of the contents of the Automated Documentation extract files that allow you to design additional reports as needed, see Creating Customized Reports.

General Automated Documentation Procedures

As the Automated Documentation facility allows you to access and organize TDF information in a wide variety of ways, advance planning allows you to obtain exactly the information that you want.

We recommend the following steps:

- Decide which reports you need to produce
- Determine what elements you want to include on the reports
- Create new extract files, if necessary, customizing the control cards to get the needed records on the extract files
- Customize the control cards for your selected reports, and submit the JCL to create the reports

JCL Provided on the Install Tape

JCL is provided as part of the CA Telon installation to create the extract files, standard reports, and the Global Cross-Reference reports. These jobs are installed in the pdsqual.INSTALL PDS.

Job for Creating Extract Files

Job JUXTRT is used to create the extract files, which is done before running the reports. JUXTRT invokes procedure TLNUXTRT.

An extract control card is used to customize the contents of the extract files. If the extract files already exist, they are deleted and new extract files are created.

Additional control cards may be included in this job to generate the standard reports after creation of the extract files. The report control cards can be used to customize the contents of the reports. As delivered, the JUXTRT and TLNUXTRT proc job will generate all standard reports selecting all members on the extract files.

Standard Reports Job

The JUXREF job, which invokes the procedure TLNUXREF, is used to generate any or all of the standard reports from existing extract files. The report control cards may also be used in this job to select specific members to appear on the reports.

Job for Creating Global Cross-Reference Reports

Job JUXXRF, which invokes the procedure TLNUXXRF, is used to generate the Global Cross-Reference Reports.

Autodoc DDNames

All DDnames used in the Automated Documentation procedures are described in the following list.

The data set names assigned to many of these DDnames are common to all CA Telon-provided procedures. These are customized to your shop defaults as part of the CA Telon installation process.

Some data set names are assigned based on values supplied on the specific PROC parameters in each job. These are described in more detail in each section. The next table lists the common DDnames.

JCL DDName	Description
STEPLIB	Library where the CA Telon load modules are located

JCL DDName	Description
TNSYSLIB	Library where the CA Telon load modules are located
TLTRACE	Used for CA Telon database call tracing. Used for debugging. Activated with TRACE=Y option
SYSOUT, SYSPRINT, and SYSUDUMP	System files used for debugging
SRTPRINT	System DD used for displaying information about internal sorts which were processed
SORTWK01, SORTWK02, SORTWK03, SORTWK04, and SORTWK05	The work files used while processing the internal sorts
TNTDD	TDF Data Administration file
TNTDF	TDF Entity file
TNTCCL	TDF Custom Code file
TNTDX	TDF SQL Information file
TNERSEL	Used for reporting the validity of the input control cards
TNEREXT	Used for reporting messages during the generation of the extract files
TNIRSEL	File containing all input control cards
TNRCCER	Custom Code Summary by Entity Report file
TNRCCNR	Custom Code Cross-Reference Report file
TNRDASR	Data Access Summary Report file
TNRSSDB	Data Access Detail Report file
TNRSSER	Data Access Detail by Entity Report file
TNXCUST	Custom Code extract file
TNXDBSEG	Data Access/Segment extract file
TNXENT	Entity extract file

Extract Files

To generate Automated Documentation more efficiently, CA Telon processes the database only once and generates three extract files. This reduces the number of I/O calls made. Two of the extract files are used to generate the five standard reports. All three files permit the creation of custom reports. These extract files are:

- Custom Code Extract (TNXCUST)—Used for the Entity Custom Code Summary and the Custom Code Cross-Reference Reports.
- Data Access/Segment Extract (TNXDBSEG)—Used for the Data Access Summary, the DBD Name Segment/File Summary and the Entity Segment/File Summary Reports.
- Entity Extract (TNXENT)—Additional information about each entity. This file is not used in creating the standard Autodoc reports.

Custom Code Extract (TNXCUST)

To produce this extract file, CA Telon examines every CA Telon statement and parameter pair on the TDF to find all references of one or more Custom Code members that can be found. For each custom code name found, CA Telon builds an extract record with information about where this name is found (Entity, Entity Header, Entity Type, CA Telon statement, and CA Telon parameter).

CA Telon searches the TNTCCL file (the TDF custom code file) to locate the actual member by this name. If this search is successful, the characters *TDF*, and additional information about this member, are added to the Extract Record. Otherwise, only the characters *PDS* are added to the Extract Record, indicating that the member does not exist or is found in a library external to the TDF.

Data Access/Segment Extract (TNXDBSEG)

To produce this extract file, CA Telon processes every data access request within a given entity. This processing involves accessing any segments that can be found. For each segment found, a search is made for any data access requests against it. CA Telon also stores a count of the number of times the segment is accessed, to verify its existence, in the extract record. At each level of access, CA Telon stores specific identifying information in the extract record.

Entity Extract (TNXENT)

The Entity Extract file records the following information about every entity accessed while producing the other two extract files:

- Full entity name
- Entity header
- Entity ID
- Entity type (values: PI, PD, SD, BD, DR, ND, SP, RD)
- Status (one-character entity type code plus X to indicate incomplete)
- Last user to update the entity
- Creation date
- Date the entity was last updated
- Time the entity was last updated
- Forty-character description of the entity

Create Extract Files

You can create new extract files by performing the following steps:

1. Examine the TLNUXTRT proc member (in the INSTALL data set). This proc creates the extract files that are used by the CA Telon Automated Documentation Facility and that you can use later in writing your own customized reports.
2. Review the parameters for the proc, and modify them, if necessary. Most of the parameters are common to all CA Telon-provided JCL. These are customized to your shop defaults as part of the CA Telon installation process. See the Installation Guide for details on these parameters, if you think you need to change them for this job.

   The parameters specific to this job are:

   - XCUST—DSN of the extract file that will hold custom code information
   - XDBSEG—DSN of the extract file that will hold Data Access information
   - XENT—DSN of the extract file that will hold information about the TDF Entities
   - SELCTL—DSN of the file that contains the control cards for the Extract program (and for any reports to be created)

3. Review and update the control cards for input to this proc. A default control card is contained in the CIXSEL member in the INSTALL data set. The control card member for this job should contain a control card specifying parameters for the creation of the Automated Documentation extract files.

   You can also add standard report control cards after the extract control card to produce the standard reports in the same job execution.

Control Card Description

Format

```
          1    1    2    2    3
....5....0....5....0....5....0....
NNNNHHHHHTT
```

Field descriptions

- **Parameter**—NNNN
  - **Field**—Extract Program Name
  - **Columns**—1-4
  - **Req/Opt**—Required
  - **Description**—Required value is REXT
- **Parameter**—HHHHH
  - **Field**—Entity Header
  - **Columns**—5-9
  - **Req/Opt**—Optional
  - **Description**
    - All TDF entities that have an entity header equal to the value entered here are selected.
    - When you enter a specific header value, only entities with that specific header are selected.
    - The asterisk (*) can be used as a trailing wildcard selector. If you want a range of headers, enter a partial header-ID followed by asterisks.
    - If the value is all spaces or all asterisks, CA Telon selects all TDF entities.

- **Parameter**—TT
  - **Field**—Entity Type
  - **Columns**—10-11
  - **Req/Opt**—Optional
  - **Description**—Values are:
    - **PD**—Panel Definition
    - **PI**—Panel Image
    - **SD**—Screen Definition
    - **BD**—Batch Definition
    - **DR**—Driver Definition
    - **RD**—Report Definition
    - **ND**—CICS Non-Terminal Definition
    - **SP**—Stored Procedure Definition

      All TDF entities that have an entity type equal to the value of this field are selected. If the value is all spaces or all asterisks, CA Telon selects all TDF entities.

Standard Reports

There are three categories of standard reports:

- Custom Code
- Data Access
- SQL Name

This section provides information about how to run these reports and samples of reports.

Produce Standard Reports

Follow the steps shown next to generate the CA Telon Automated Documentation standard reports.

To generate standard reports

1. Choose the appropriate JCL to run:
   - If you want to produce new extract files before the standard reports, use JCL member JUXTRT and PROC member TLNUXTRT
   - If the extract files you want to use have already been created, use JCL member JUXREF PRT and PROC member TLNUXREF
2. Review the parameters to the chosen PROC and modify them, if necessary

   The parameters specific to this job are as follows:
   - **XCUST**—DSN of the extract file that holds custom code information.
   - **XDBSEG**—DSN of the extract file that holds Data Access information.
   - **XENT**—DSN of the extract file that holds information about the TDF Entities.
   - **SELCTL**—DSN of the file that contains the control cards. For TLNUXTRT, the file contains both Extract control card and the standard report control cards. For TLNUXREF, the file contains only the standard Report control cards.
3. Review and update the control cards for input to this PROC. Default control cards for TLNUXTRT are contained in member CIXSEL in the INSTALL data set. Default control cards for TLNUXREF are contained in member CIRSEL in the INSTALL data set.

Control Card Description

The following types of control cards are available, according to the type of standard report specified:

- Custom Code Reports
- Data Access
- Data Access Detail
- SQL Table Name Cross-Reference

Custom Code Reports Control Card

This control card is used to produce the following reports:

- Custom Code Summary by Entity Report
- Custom Code Cross-Reference Report

Format

```
          1    1    2    2    3       6    6    7    7    8
....5....0....5....0....5....0....   0....5....0....5....0
NNNNHHHHHTTsssssspppppppp                  CCCCCCCC
```

Field descriptions

- **Parameter**—NNNN
  - **Field**—Report Name
  - **Columns**—1-4
  - **Req/Opt**—Required
  - **Description**—Values are:
    - **CCER**—Custom Code Summary by Entity Report
    - **CCNR**—Custom Code Cross-Reference
- **Parameter**—HHHHH
  - **Field**—Entity Header
  - **Columns**—5-9
  - **Req/Opt**—Optional
  - **Description**:

    All TDF entities that have an entity header equal to the value entered here are selected. When you enter a specific header value, only entities with that specific header are selected. The asterisk (*) can be used as a trailing wildcard selector. If you want a range of headers, enter a partial header-ID followed by asterisks. If the value is all spaces or all asterisks, then CA Telon selects all TDF entities.

- **Parameter**—TT
  - **Field**—Entity Type
  - **Columns**—10-11
  - **Req/Opt**—Optional
  - **Description**—Values are:
    - **PD**—Panel Definition
    - **PI**—Panel Image
    - **SD**—Screen Definition
    - **BD**—Batch Definition
    - **DR**—Driver Definition
    - **RD**—Report Definition
    - **ND**—CICS Non-Terminal Definition
    - **SP**—Stored Procedure Definition
- **Parameter**—sssssssppppppp
  - **Field**—Selection Field (occurs four times)
  - **Columns**
    - **12-25**
    - **26-39**
    - **40-53**
    - **54-67**
  - **Req/Opt**—Optional
  - **Description**—This field consists of:
    - CA Telon statement name.
    - Parameter (length 7).
    - Within the TDF, a program is defined in a series of statements. Each of the statements has a set of valid parameters.
    - By entering values in these fields, you can control where the search is made for different parts of a CA Telon TDF Program Definition.
    - For a list of valid CA Telon statements and their parameters, see the *Programming Concepts Guide*.
    - If all four Selection Fields are left blank, or any one of the four is filled with asterisks (wildcard), CA Telon assumes that every statement record should be searched.

- **Parameter**—CCCCCCCC
  - **Field**—Custom Code Name
  - **Columns**—68-75
  - **Req/Opt**—Optional
  - **Description**—Name of the custom code member for which to search.
  - To see a report showing every place where a specific custom code member is referenced, enter the name of the custom code member in this field.

    You can use asterisks to indicate wildcard selections in this field. A generic name, followed by asterisks, tells CA Telon to search for all custom code members beginning with the generic name.

Data Access Control Card

Format

```
          1    1    2    2    3
....5....0....5....0....5....0....
NNNN
```

Field description for Parameter—NNNN

- **Field**—Report Name
- **Columns**—1-4
- **Req/Opt**—Required
- **Description**—Required value is DASR for Data Access Summary Report.
  - No selection criteria is required.
  - If the control card is present, the report is generated.

Data Access Detail Control Card

Format

```
          1         2         3         4         5
....5....0....5....0....5....0....5....0....5....0
NNNNHHHHHTTAAIDDDDDDDDLLLPPPPPPPPPPPPMSSSSSSSSCC
```

Field Descriptions

Parameter—NNNN

- **Field**—Report Name
- **Columns**—1-4
- **Req/Opt**—Required
- **Description**—Values are:
  - SSDB —Data Access Detail Report
  - SSER —Data Access Detail by Entity Report

Parameter—HHHHH

- **Field**—Entity Header
- **Columns**—5-9
- **Req/Opt**—Optional
- **Description**
  - All TDF entities that have an entity header equal to the value entered here are selected.
  - When you enter a specific header value, only entities with that specific header are selected.
  - The asterisk (*) can be used as a wildcard selector. If you want a range of headers, enter a partial header-ID followed by asterisks.
  - If the value is all spaces or all asterisks, CA Telon selects all TDF entities.

Parameter—TT

- **Field**—Entity Type
- **Columns**—10-11
- **Req/Opt**—Optional
- **Description**—Values are:
  - PD —Panel Definition
  - PI —Panel Image
  - SD —Screen Definition
  - BD —Batch Definition
  - DR —Driver Definition
  - RD —Report Definition
  - ND —CICS Non-Terminal
  - SP —Stored Procedure Definition Definition
  - All TDF entities that have an entity type equal to the value of this field are selected.
  - If the value is all spaces or all asterisks, CA Telon selects all TDF entities with any kind of file or data access.

Parameter—AA.

- **Field**—Access Indicator.
- **Columns**—12-13.
- **Req/Opt**—Required.
- **Description**—Values are:
  - ON —Select DL/I databases (D) and teleprocessing PCBs (T)
  - DB —Select DL/I databases only (D)
  - TP —Select Teleprocessing PCBs only
  - BA —Select Sequential and VSAM data sets
  - DS —Select Sequential and VSAM data sets
  - VS —Select VSAM only
  - SQ —Select Sequential only
  - TB —Select SQL Tables and Joins

  All TDF program definitions containing this type of access are selected.

Parameter—I

- **Field**—DBD/PSB Indicator

  Columns—14

- **Req/Opt**—Optional
- **Description**—Value is M. Refer to the next three fields to define the report criteria to specific DBDs or PSBs.

  If this field is left blank, or filled with asterisks, the report selection criteria ignores the next three fields.

Parameter—DDDDDDDD

- **Field**—DBD Name
- **Columns**—15-22
- **Req/Opt**—Optional
- **Description:**
  - CA Telon selects all TDF entities that match the specific or generic DBD name.
  - When you enter a specific eight-character DBD name, only entities referencing that specific DBD are selected.
  - The asterisk (*) can be used as an imbedded or a trailing wildcard selector. If you want a range of DBDs, enter a partial DBD name either with asterisks in the name, or followed by asterisks.
  - The space is also used as a trailing wildcard selector.
  - If the value is all spaces or all asterisks, CA Telon selects all TDF entities with any sort of DBD reference.

Parameter—LLL

- **Field**—Logical Connector
- **Columns**—23-25
- **Req/Opt**—Optional
- **Description**—Values are:
  - AND —Indicates a TDF entity should be selected only when both the Generic DBD Name and the Generic PCB Name match
  - OR —Indicates a TDF entity should be selected if either the Generic DBD Name or the Generic PCB Name, or both, match
  - Blank —Same as AND

  This field serves as a logical connector between the previous field and the next field.

Parameter—PPPPPPPPPPP

- **Field**—Generic PCB Name
- **Columns**—26-37
- **Req/Opt**—Optional
- **Description**—CA Telon selects all TDF entitie s that match the specific or generic PCB name.
  - When you enter a specific 12-character PCB name, only entities referencing that specific PCB are selected.
  - The asterisk (*) can be used as an imbedded or a trailing wildcard selector. If you want a range of PCBs, enter a partial PCB name either with asterisks in the name, or followed by asterisks.
  - The space is also used as a trailing wildcard selector.
  - If the value is all spaces or all asterisks, CA Telon selects all TDF entities with any sort of PCB reference.

Parameter—M

- **Field**—Segment Mask Indicator
- **Columns**—38
- **Req/Opt**—Optional
- **Description**—Valid value R. Refer to the next field to refine the report criteria to specific Segment Names

  If this field is left blank, or filled with asterisks, the report selection criteria ignores the next field.

Parameter—SSSSSSSS

- **Field**—Segment Name
- **Columns**—39-46
- **Req/Opt**—Optional
- **Description**—CA Telon selects all TDF entities that match the specific or generic Segment name.
  - When you enter a specific eight-character Segment name, only entities referencing that specific Segment are selected.
  - The asterisk (*) can be used as an imbedded or a trailing wildcard selector. If you want a range of Segments, enter a partial Segment name either with asterisks in the name, or followed by asterisks.
  - The space is also used as a trailing wildcard selector.
  - If the value is all spaces or all asterisks, CA Telon selects all TDF entities.

Parameter—CC

- **Field**—Call Indicator
- **Columns**—47-48
- **Req/Opt**—Required
- **Description**—Values are:
  - **AU**—AUTOEXEC only
  - **B**—BROWSE & SPBROWSE
  - **B***—BROWSE & SPBROWSE
  - **BR**—BROWSE & SPBROWSE
  - **C**—all CREATEs
  - **C***—all CREATEs
  - **CA**—AUTOEXEC CREATE
  - **CU**—USEREXEC CREATE
  - **D**—all DELETEs
  - **D***—all DELETEs
  - **DA**—AUTOEXEC DELETE
  - **DU**—USEREXEC DELETE
  - **E**—ERASE
  - **E***—ERASE
  - **ER**—ERASE
  - **I**—INREAD
  - **I***—INREAD
  - **IN**—INREAD
  - **OI**—OIREAD
  - **OU**—OUTREAD
  - **R**—READ & READNEXT & SPRDNEXT
  - **R***—READ & READNEXT & SPRDNEXT
  - **RD**—READ
  - **RN**—READNEXT & SPRDNEXT
  - **T**—TRANSACT & SPTRNACT
  - **T***—TRANSACT & SPTRNACT
  - **TR**—TRANSACT & SPTRNACT
  - **U**—all UPDATEs

- **U***—all UPDATEs
- **UA**—AUTOEXEC UPDATE
- **US**—USEREXEC only
- **UU**—USEREXEC UPDATE; indicates the request type for which data access is selected .

SQL Cross-Reference Reports Control Card

Format

```
          1    1    2    2    3
....5....0....5....0....5....0....
NNNN
```

Field description for Parameter—NNNN

- **Field**—Report Name
- **Columns**—1-4
- **Req/Opt**—Required
- **Description**—Values are:
  - **SQL2**—SQL Table Name Cross-Reference Report
  - **SQLT**—CA Telon Name Cross-Reference Report
  - No selection criteria is required
  - If the control card is present, the requested report is generated

Examples—General Description

The Automated Documentation Facility reports are:

- Custom Code Cross-Reference
- Custom Code by Entity
- Data Access Summary
- Data Access Detail
- Data Access Detail by Entity
- SQL Table Name Cross-Reference
- CA Telon Name Cross-Reference

Descriptions of each of these reports, including field descriptions and samples, are provided next.

Custom Code by Entity/Custom Code Cross-Reference

These reports give a list of where custom code is used within an application to identify where common custom code components are used across systems. The reports list those custom code member names that are referenced in the user-designed programs under the TDF.

For each custom code member name referenced, CA Telon searches the TDF database for the existence of the actual member. If the search is successful, CA Telon retrieves and prints information about that member. Otherwise, CA Telon prints the characters PDS or UNK next to the member name. PDS means the code is kept on a library external to the TDF. UNK means CA Telon is unable to identify where the code is kept.

The sort order for the *Custom Code Summary by Entity Report* is: entity, CA Telon statement, CA Telon parameter, and custom Code member name.

The sort order for the *Custom Code Cross-Reference Report* is: custom code member, entity, CA Telon statement, and CA Telon parameter.

Sample report

A sample of the Custom Code Summary by Entity report is shown next.

```
6/16/05                              C u s t o m  C o d e  S u m m a r y  b y  E n t i t y                    Page  1
10:19:26


SELECTION CONTROLS ==< ENTITIES:                    STATEMENT(S):  *ALL*        PARAMETER(S): *ALL*         Code Name: *ALL* :
                       <------1------>
                       * Header *ALL*               <---------------------2---------------------->       <--------3------->
                       * All Types
                                                                    <-5-->    <----------------------6---------------------->

<------------------------4-------------------------->  Source   ****** Additional Information for TDF Members Only ******
Used by Entity  Statement -Parameter  Custom Code Name  (PDS/TDF)  Where Size in Lines  Date Last Updated  Last Update User
_____________________________________________________________________________________________________________________________
BDO1.BD         BATCH     -GETTRAN    GETTRAN            TDF                2          08/03/05          SMALL2
                          -INIT       INIT               TDF                4          08/03/05          SMALL2
                          -PRCTRAN    PRCTRAN            TDF                2          08/03/05          SMALL2
                          -REMARKS    REMARKS            TDF                1          08/03/05          SMALL2
                          -TERM       TERM               TDF                2          08/03/05          SMALL2
                RECORD    -COPY       SEQCOPY        PDS
                          -COPY       VSAMCOPY       PDS
_____________________________________________________________________________________________________________________________
DRVR.DR         DRIVER    -INIT       INIT               TDF                1          05/03/05          SMALL1
                          -REMARKS    REMARKS            TDF                1          05/03/05          SMALL1
                          -TERM       TERM               TDF                1          05/03/05          SMALL1
                          -XFER       XFER               TDF          1           05/03/05          SMALL1
_____________________________________________________________________________________________________________________________
PDSL.PD         FIELD     -SCONSIS    SCONPIPD       PDS
                SEGLOOP   -ICUSTOM    ICUSTPD2       PDS

                          -OCUST1     OCUSTPD1       PDS

                          -OCUST2     OCUSTPD2       PDS
                          -OCUST3     OCUSTPD3       PDS
_____________________________________________________________________________________________________________________________
PIDG.PD         GROUP     -FMTCUST    FMTCUST3       PDS

                          -FMTCUST    TPFMTCST       PDS

_____________________________________________________________________________________________________________________________
PIG2.PD         GROUP     -FMTCUST    FMT01          PDS

                          -FMTCUST    FMT2           PDS

                          -FMTCUST    FMT3           PDS

_____________________________________________________________________________________________________________________________
PIPD.PD         FIELD     -SCONSIS    SCONPIPD       PDS

_____________________________________________________________________________________________________________________________
REPT.RD         REPORT    -LINKWKA    LINKWKA            TDF                1          05/03/05          SMALL1
                          -OINIT1     OINIT1             TDF                1          05/03/05          SMALL1
                          -OINIT2     OINIT2             TDF                1          05/03/05          SMALL1
                          -OUTTERM    OUTTERM            TDF                1          05/03/05          SMALL1
```

3/14/05 C u s t o m C o d e C r o s s R e f e r e n c e
11:27:15

SELECTION CONTROLS ==< ENTITIES: STATEMENT(S): *ALL* PARAMETER(S): *ALL* Code Name: *ALL* :
<------1------>
* Header *ALL* <----------2----------> <-------3------>
* All Types

<-------4------->				<-5->	<-------6------->		
Custom Code Name	Used by Entity	Statement –Parameter	(PDS/TDF)	Source Where	Size in Lines	Date Last Updated	Last Update User
Y	TRCDAD.SD	IMSPGM –MSGCOPY	PDS				
	TRCVLI.SD	IMSPGM –MSGCOPY	PDS				
	TRTLAD.SD	IMSPGM –MSGCOPY	PDS				
CMKERR	TRCVTA.PD	SEGLOOP –ICUSTOM	TDF	SD	47	02/20/05	BATCHIMP
CONSIS	TRCCDA.SD	SCREEN –CONSIS	TDF		9	09/24/05	HARRIG1
	TRCMBS.SD	SCREEN –CONSIS	TDF		15	06/14/05	BATCHIMP
	TRCTSA.SD	SCREEN –CONSIS	TDF		9	09/24/05	HARRIG1
	TRDON2.SD	SCREEN –CONSIS	TDF		80	08/27/05	CHRISO1
	TRMCMB.SD	SCREEN –CONSIS	TDF		15	05/09/05	BATCHIMP
	TRLORN.SD	SCREEN –CONSIS	PDS				
	TRMENU.SD	SCREEN –CONSIS	TDF		15	03/03/05	HARRIS1
	TRPMNU.SD	SCREEN –CONSIS	TDF		17	09/04/05	BATCHIMP
	TRRBOS.SD	SCREEN –CONSIS	TDF		15	01/13/05	BURROUG
CURSCUS	TRCVTA.SD	SEGLOOP –CURSUS	TDF		14	02/20/05	BATCHIMP
	TRTASK.SD	SEGLOOP –CURSUS	TDF		14	02/20/05	BATCHIMP
DARKFLD	TRBQ2O.BD	BATCH –WKAREA	TDF		15	11/25/05	QUINN
DELETE	TRCVTA.PD	FIELD –SCONSIS	TDF	SD	12	02/20/05	BATCHIMP
	TRTASK.PD	FIELD –SCONSIS	TDF	SD	10	08/21/05	BATCHIMP
DUPLICATE	TRCVTA.PD	FIELD –SCONSIS	TDF	SD	35	02/20/05	BATCHIMP
	TRTASK.PD	FIELD –SCONSIS	TDF	SD	33	08/21/05	BATCHIMP
ERRFORCE	TRCVTA.SD	FIELD –SCONSIS	TDF		16	02/20/05	BATCHIMP
	TRTASK.SD	FIELD –SCONSIS	TDF		16	08/21/05	BATCHIMP
EXTRACT	TRBQ2O.BD	SEGMENT –COPY	PDS				
	TRCM2O.BD	SEGMENT –COPY	PDS				

Selection controls

The following numbers correspond to the numbers on the reports.

- **Entities**—*Entity* refers to a program, in whole or in part, that was designed in the TDF. The entity name is made up of three distinct parts: header (1 to 5 characters), ID (1 to 5 characters), and type (2 characters, prefixed with a period). A complete CA Telon program consists of one or more distinct entities all having the same header and ID: one and only one entity of type SD, DR, RD, BD, SP, or ND; and, possibly, one each of type PD and PI.

  The second line of this section of the report page displays the entity header being selected by using the control card, or the characters "*ALL*", meaning all entity headers.

  The third line of this section of the report page displays the entity type being selected by using the control card, or the characters "All Types," meaning all entity types.

- **Statements/Parameters**—*Statement* refers to a specific component of a CA Telon program. *Parameter* refers to a specific CA Telon part of that component. If a specific value is not selected by using the control card, then the characters "*ALL*" appear in the report and all values are selected.

  Up to four statements can be selected on the control card, each statement with one parameter. The report lists all statements from the control card followed by all parameters. The first statement value must be thought of in conjunction with the first parameter value and no other.

- **Code Name**—Code Name refers to the name of a custom code member.

  For example, if you were interested in identifying all programs that specify a PGMCUST of H100T, the Code Name field would contain the value H100T. If you do not provide a specific value, the characters "*ALL*" appear in the report and all values are selected.

- **Detail Information**—The first four columns correspond to the four fields of the selection control card. The entity name is listed in the format HHIIII.TT (where HH = header, IIII = ID, and TT = type).

- **Source**—*Source* refers to the location of the custom code. The only possible values for this column are PDS, UNK, and TDF.

  If the custom code member is found within the TDF, the characters TDF are displayed in this field. If CA Telon cannot identify the source language, it displays UNK. Otherwise, the characters PDS appear.

  The characters PDS are left-justified while the characters TDF are right-justified within the column.

- **TDF Information**—The remaining columns are only filled in when the source is TDF.
  - *Where* refers to the parent entity of the custom code when the current entity is of type PD.
  - *Size in Lines* refers to the number of lines of source code actually in the custom code member.
  - *Date Last Updated* refers to the last time a user made changes to the custom code member.
  - *Last Update User* indicates the TDF logon ID of last user to change the custom code member. Last Update User has a value of BATCHIMP if the member was imported by using JCL.

Examples—Data Access Reports

There are three kinds of Data Access reports:

- Data Access Summary
- Data Access Detail
- Data Access Detail by Entity

Data Access Summary Report

The Data Access Summary Report summarizes the use of databases and files referenced in application programs by database or file name.

The report lists all data access by DBD name, SQL table TLNNAME, VSAM or sequential file name, CICS queue or journal name, or TPPCB name referenced by user-designed programs.

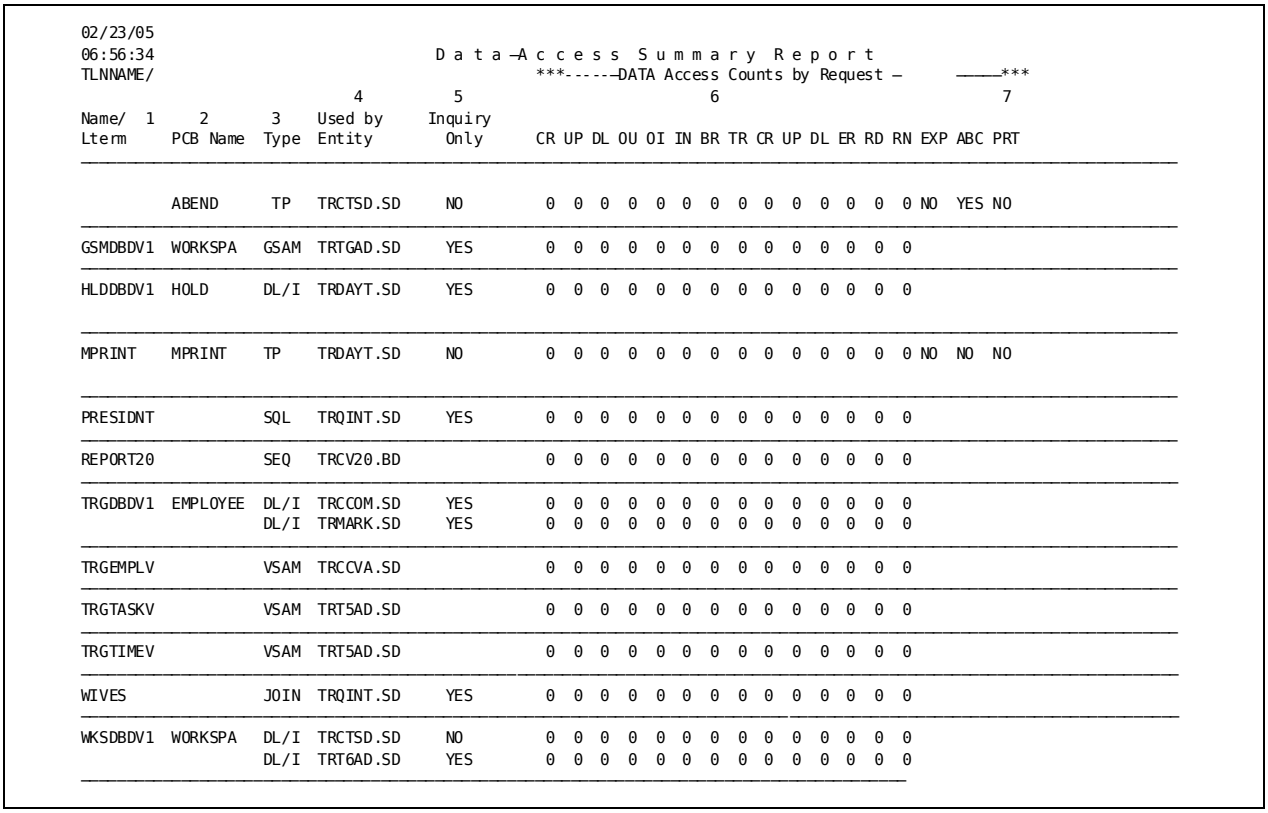
For each data access referenced, the report indicates:

- *Type* of data access
- *Entity* that references it
- Whether the access is of an *inquiry nature only* (DL/I only)
- Number of data access *calls made against* it
- *Values* of the TPPCB parameters (if data access type is TP)

The sort order for this report is DBD name, PCB name, and entity. Currently, the only selection for this report is to indicate whether the report should be generated, that is, all data accesses are currently listed.

Sample report

A sample of the Data Access Summary report is shown next.



```
02/23/05
06:56:34                          D a t a - A c c e s s  S u m m a r y  R e p o r t
TLNNAME/                                    ***------DATA Access Counts by Request -      -----***
                          4         5                          6                                7
Name/  1     2       3    Used by     Inquiry
Lterm     PCB Name  Type  Entity        Only      CR UP DL OU OI IN BR TR CR UP DL ER RD RN EXP ABC PRT
_____________________________________________________________________________________________________

          ABEND      TP   TRCTSD.SD     NO        0  0  0  0  0  0  0  0  0  0  0  0  0  0 NO  YES NO
_____________________________________________________________________________________________________
GSMDBDV1  WORKSPA   GSAM  TRTGAD.SD     YES       0  0  0  0  0  0  0  0  0  0  0  0  0  0
_____________________________________________________________________________________________________
HLDDBDV1  HOLD      DL/I  TRDAYT.SD     YES       0  0  0  0  0  0  0  0  0  0  0  0  0  0

_____________________________________________________________________________________________________
MPRINT    MPRINT    TP    TRDAYT.SD     NO        0  0  0  0  0  0  0  0  0  0  0  0  0  0 NO  NO  NO

_____________________________________________________________________________________________________
PRESIDNT            SQL   TRQINT.SD     YES       0  0  0  0  0  0  0  0  0  0  0  0  0  0
_____________________________________________________________________________________________________
REPORT20            SEQ   TRCV20.BD               0  0  0  0  0  0  0  0  0  0  0  0  0  0
_____________________________________________________________________________________________________
TRGDBDV1  EMPLOYEE  DL/I  TRCCOM.SD     YES       0  0  0  0  0  0  0  0  0  0  0  0  0  0
                    DL/I  TRMARK.SD     YES       0  0  0  0  0  0  0  0  0  0  0  0  0  0
_____________________________________________________________________________________________________
TRGEMPLV            VSAM  TRCCVA.SD               0  0  0  0  0  0  0  0  0  0  0  0  0  0
_____________________________________________________________________________________________________
TRGTASKV            VSAM  TRT5AD.SD               0  0  0  0  0  0  0  0  0  0  0  0  0  0
_____________________________________________________________________________________________________
TRGTIMEV            VSAM  TRT5AD.SD               0  0  0  0  0  0  0  0  0  0  0  0  0  0
_____________________________________________________________________________________________________
WIVES               JOIN  TRQINT.SD     YES       0  0  0  0  0  0  0  0  0  0  0  0  0  0
_____________________________________________________________________________________________________
WKSDBDV1  WORKSPA   DL/I  TRCTSD.SD     NO        0  0  0  0  0  0  0  0  0  0  0  0  0  0
                    DL/I  TRT6AD.SD     YES       0  0  0  0  0  0  0  0  0  0  0  0  0  0
_____________________________________________________________________________________
```

Detail information

The following numbers correspond to the numbers on the report.

1. **Name/LTerm**—The DBD name, SQL table TLNNAME, VSAM or sequential file name, CICS queue or journal name, or TPPCB name (LTerm) of the data access being reported.
2. **PCB Name**—The name of the program control block (PCB) found under this data access (DL/I only).
3. **Type**—The type of data access being reported. The reported values are DL/I (database), SQL (SQL Table), JOIN (SQL Join), TP (teleprocessing program control block or TPPCB), SEQ (sequential file), VSAM (virtual sequential access method file), and GSAM.
4. **Used by Entity**—A program, in whole or in part, designed under the TDF, as described in detail in the preceding report description.
5. **Inquiry Only**—This column is only valid for data accesses of type DL/I (database) or TP (teleprocessing control block or TPPCB). It indicates whether or not calls made to this data access are of an inquiry nature only. If this field is blank, no calls were made to this data access. This field is blank for non-DL/I data accesses.
6. **Data Access Counts by Request**—This section of the report page is for summarizing the data accesses for each DB, Table, or file. The request types are abbreviated as follows:
   - AUTOEXEC
     - BR (browse —read in a loop on the output side)
     - CR (create)
     - DL (delete)
     - IN (inread —read on the input side)
     - OI (oiread —read on both the input and output side)
     - OU (outread —read on the output side)
     - TR (transact)
     - UP (update)
   - USER EXEC
     - CR (create)
     - DL (delete)
     - ER (erase)

- RD (read)
- RN (read next)
- UP (update)

7. **TPPCB**—This section of the report page is specifically for data access of type TP (teleprocessing control block or TPPCB). This section is divided into three columns: EXP (express), ABC (abcall), PRT (print). If the data access is not of type TP, these fields are blank.

Data Access Detail Report

This report summarizes the data accesses for a system and identifies where and how database segments or files are used. It lists those segments (or filenames) that are referenced by user-designed programs found in the TDF.

For each segment or file referenced, the report prints:

- Entity that references the segment (or file)
- Name of any secondary index used to access the segment (or file)
- Name of the data key for the segment (or file)
- Type of processing associated with the segment (or file)
- Whether I/O statements are generated automatically
- Number of data accesses against the segment (or file) by type
- Number of SEGEDITs made against the segment (or file)

The sort order for the DBD Name Segment/File Summary Report is DBD name, PCB name, segment name, and entity.

Sample report

A sample of the Data Access Detail report is shown next.

```
   02/24/05

    13:17:18                          D a t a - A c c e s s  D e t a i l  R e p o r t

    Selection Controls ==< Generic Names:          Segments:                    Calls:            Entities: 4
                           * All Types    1         * All Types  2                * All Types 3
* Header *ALL*
                                                                                                    * All Types
    Table / 5              Row /                                                    8                          9
    Join /                 Record /                                  ***-------Data Access Counts by Request ————————***
    File /                 Segment                6             7                AUTOEXEC                USEREXEC  SEGEDIT
    DBD Name   PCB Name    Name         Entity  Procseq          IMSkey     CR UP DL OU OI IN BR TR CR UP DL ER  D RN Count
    ———————————————————————————————————————————————————————————————————————————————————————————————————————————————————
    TRGDBD01   EMPLOYEE    TRGTIME      TRT511.BD              TRGTIKEY    0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
                                        TRT512.BD              TRGTIKEY    0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
                                        TRUPDT.SD              TRGTIKEY    0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
    ———————————————————————————————————————————————————————————————————————————————————————————————————————————————————
    TRGDBD02   HELP        TRGEMPLV     TRNKME.SD                          0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
                                        TRCVTA.SD                          0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
                                        TRCTST.SD                          0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
    ———————————————————————————————————————————————————————————————————————————————————————————————————————————————————
    TRGTASKV               TRGTASKV     TRCVTA.SD                          0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
    ———————————————————————————————————————————————————————————————————————————————————————————————————————————————————
    TRGTIMEV               TRGTIMEV     TRCVTA.SD                          0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
    ———————————————————————————————————————————————————————————————————————————————————————————————————————————————————
    WKSDB001   WORKSPA                  TRHIG1.BD                          0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
                           XFERA        TRDRUP.SD              WKSRTKEY    0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
                           XFERB                               *NOKEY      0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
                           XFERC                               *NOKEY      0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
                           XFERWKA      TRACDS.SD              WKSRTKEY    0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
                                        TRBQAD.SD              WKSRTKEY    0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
                                        TRBQUP.SD              WKSRTKEY    0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
                                        TRDISP.SD              WKSRTKEY    0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
                                        TRDISP.SD              WKSRTKEY    0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
                                        TRDCN2.SD              WKSRTKEY    0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
                                        TRDOUG.SD              WKSRTKEY    0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
                                        TRDRVR.RD              WKSRTKEY    0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
                                        TRDRVI.RD              WKSRTKEY    0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
                                        TRJRB1.SD              WKSRTKEY    0  0  0  0  0  0  0  0  0  0  0  0  0 0 0   0
    ———————————————————————————————————————————————————————————————————————————————————————————————————————————————————
```

Selection controls

The following numbers correspond to the numbers on the report.

1. **Generic Names**—The term "generic" refers to the option of specifying wildcard characters (spaces or asterisks) to replace other characters in the name. When testing is made as to whether the name being read matches the generic name, wildcard characters are always indicated as a match.

   The first line of this section has one of two values: "All Types" or "Only:." If the value is "All Types" then the remaining three lines are blank. Otherwise, the second line contains the generic DBD name (or file name), the fourth line contains the generic PCB name, and the third line contains the logical operator used to connect these two generic names (either OR or AND).

2. **Segments**—The first line of this section has "All Types" to match the user-supplied value.

3. **Calls**—This term refers to the type of User Exec (User I/O) calls which are to be selected.

4. **Entities**—The term "entity" refers to a program that has been designed using CA Telon.

5. **Detail Information**—The first four columns correspond with the four columns of the selection controls. Values listed in these columns are the actual values found.

6. **Procseq**—This column refers to the name of a secondary index that is used to access this database.

7. **IMSkey**—This column refers to the name of the key for the segment as defined to DL/I in the DBD. This is also the SSA key name.

8. **Data Access Counts by Request**—The term "data access" refers to the input/output operations which are able to be called from anywhere custom code members can be placed. This column indicated the number of such I/O operations requested.

9. **Segedit Count**—The term "segedit" refers to code that accesses a segment to verify that the segment is found. This column indicates the number of SEGEDITs that were requested against this segment.

Data Access Detail by Entity Report

This report summarizes the data accesses for a system and identifies where and how database segments or files are used. It lists those segments (or filenames) that are referenced by user-designed programs found in the TDF.

For each segment or file referenced, the report prints the following:

- Entity that references the segment (or file)
- Name of any secondary index used to access the segment (or file)
- Name of the data key for the segment (or file)
- Type of processing associated with the segment (or file)
- Whether I/O statements are generated automatically
- Number of data accesses against the segment (or file) by type
- Number of SEGEDITs made against the segment (or file)

The sort order for the Entity Segment/File Summary Report is entity, BD name, PCB name, and segment name.

Sample report

A sample of the Data Access Detail by Entity report is shown next.

```
02/24/05
13:17:15          D a t a - A c c e s s  D e t a i l  b y  E n t i t y  R e p o r t
Selection Controls ==< Entities:
                        * Header *ALL*  3      Generic Names:            Segments:             Calls
                        * All Types             * All Types  1             * All Types 2     * All Types* 4
Table /               Row /                                                    8
Join /    5             Record /                    ***-------Data Access Counts by Request ————————***
File /                  Segment                    6         7                    AUTOEXEC     USEREXEC    SEGEDIT  9
Entity     DBD Name     PCB Name     Name       Procseq     IMSkey     CR UP DL OU OI IN BR TR CR UP DL ER RD RN Count
————————————————————————————————————————————————————————————————————————————————————————————————————————————————

TRCTSA.SD  MFRINT       MPPRINT                                       0  0  0  0  0  0  0  0  0  0  0  0  0  0   0
           TRGDBDV1     EMPLOYEE     TRGEMOK                TRGEMKEY      0  0  0  0  0  0  0  0  0  0  0  0  0   0
                                     TRGTASK                TRGTAKEY      0  0  0  0  0  0  0  0  0  0  0  0  0   0
                                     TRGTIME                TRGTIKEY      0  0  0  0  0  0  0  0  0  0  0  0  0   0
           WKSDBDO1     WORKSPA      XFERWKA                WKSRTKEY      0  0  0  0  0  0  0  0  0  0  0  0  0   0
                                     XFERWKB                              0  0  0  0  0  0  0  0  0  0  0  0  0   0
                                     XFERWKC                              0  0  0  0  0  0  0  0  0  0  0  0  0   0
————————————————————————————————————————————————————————————————————————————————————————————————————————————————
TRCVLI.SD  TRGEMPLV                  TRGEMPLV                             0  0  0  0  0  0  0  0  0  0  0  0  0 0   0
                                     TRGTASKV                             0  0  0  0  0  0  0  0  0  0  0  0  0   0
                                     TRGTIMEV                             0  0  0  0  0  0  0  0  0  0  0  0  0   0
————————————————————————————————————————————————————————————————————————————————————————————————————————————————
TRDISP.SD                ABEND                                        0  0  0  0  0  0  0  0  0  0  0  0  0  0   0
                         LPRINT      LTERM                 HLDRTKEY      0  0  0  0  0  0  0  0  0  0  0  0  0   0
           HLDBDO1       HOLD        XFERA                                0  0  0  0  0  0  0  0  0  0  0  0  0   0
                                     XFERB                                0  0  0  0  0  0  0  0  0  0  0  0  0   0
                                     XFERC                                0  0  0  0  0  0  0  0  0  0  0  0  0   0
           HLPDBDO1      HELP        HELP                  HLPRTKEY      0  0  0  0  0  0  0  0  0  0  0  0  0   0
           MPRINT        MPRINT                                           0  0  0  0  0  0  0  0  0  0  0  0  0   0
           TRGDBDV1      EMPLOYEE    TRGEMPL               TRGEMKEY      0  0  0  0  0  0  0  0  0  0  0  0  0   0
                                     TRGTIME               TRGTIKEY      0  0  0  0  0  0  0  0  0  0  0  0  0   0
           WKSDBDO1      WORKSPA     XFERWKA               WKSRTKEY      0  0  0  0  0  0  0  0  0  0  0  0  0   0

                                     XFERWKB
                                     XFERWKC                              0  0  0  0  0  0  0  0  0  0  0  0  0   0
————————————————————————————————————————————————————————————————————————————————————————————————————————————————
```

Selection Controls

The following numbers correspond to the numbers in the report.

1. **Generic Names**—Wildcard characters (spaces or asterisks) to replace other characters in the name. When testing is made as to whether the name being read matches the generic name, wildcard characters are always indicated as a match.

   The first line of this section has one of two values: "All Types" or "Only." If the value is "All Types" then the remaining three lines are blank. Otherwise, the second line contains the generic DBD name (or filename), the fourth line contains the generic PCB name, and the third line contains the logical operator used to connect these two generic names (either OR or AND).

2. **Segments**—The first line of this section has one of two values: "All Types" or "Roots Only." The value "Roots Only" indicates that only those segments which are root segments should be selected. The second line only displays when a value other than all asterisks or all spaces is put in the corresponding field in the input control card.

   When this line is displayed, it has the characters "Generic Name:" followed by the generic segment name that was in the input control card.

3. **Entities**—Refers to a program that has been designed under the TDF, as described in a previous field explanation. The second line in this section of the report page displays the entity header being selected, or the characters *ALL*. The third line of this section of the report page displays the entity type being selected by using the control card, or the characters "All Types."

4. **Calls**—Refers to the type of data access calls that are to be selected. User Exec Update calls are also selected when Create calls are requested.

5. **Detail Information**—The first four columns correspond to the four columns of the control cards. The entity name is listed in this section in the following format: HHIII.TT where:
   - HH = Header
   - IIII = ID
   - TT = Type
   - **Procseq**—This column refers to the name of a secondary index that is used to access this database.
6. **IMSkey**—This column refers to the name of the data key for the segment as defined to DL/I in the DBD. This is also the SSA key name.
7. **Data Access Counts by Request**—Refers to the input/output operations which are able to be called from anywhere custom code members can be placed. These columns indicate the number of such I/O operations requested.
8. **Segedit Count**—Refers to code that accesses a segment to determine whether the segment is found. This column indicates the number of SEGEDITs that were requested against this segment.

Examples—SQL Name Reports

This section provides examples and descriptions of the SQL Name Reports.

SQL Table Name Cross-Reference/CA Telon Name Cross-Reference

These reports provide a link between the eight-byte CA Telon name used to identify SQL objects and the fully-qualified SQL Table and Join names.

The Data Access Summary and Detail Reports refer to the tables by the eight-byte CA Telon Name. This report can be used to cross-reference the SQL Table name with the CA Telon name.

Each SQL object referenced by the Data Access Reports is represented and listed on the same line as the eight-byte CA Telon name, the fully-qualified SQL Table or Join name, and the Type (Table or Join).

The SQL Table Name Cross-Reference is sorted on SQL QUAL.NAME and the CA Telon Name Cross-Reference Report is sorted on TLNNAME.

Sample report

A sample of the SQL Table Name Cross-Reference report is shown next.

```
11/12/04            SQL TABLE NAME CROSS-REFERENCE REPORT         PAGE 1

16:05:34


TLNNAME                                                    QUAL.NAME
 TYPE
_____________________________________________________________________________________________

JOIN1                                                      DX.JOIN1
 JOIN


DXTABLE1                                                   DX.TABLE1
 TABLE

DXTABLE2                                                   DX.TABLE2
 TABLE


TRGEMPLX                                                   EMPL.TRGEMPL
 TABLE

DXEMPL2T                                                   TELON.DXSQLT
 TABLE

EDSTUDNT                                                   TELON.EDSTUDNT.
 TABLE

EMPLTASK                                                   TELON.EMPL_TASK_JOIN
 TABLE

TRGEMPL                                                    TELON.TRP1EMPL
 TABLE
```

Sample report

A sample of the CA Telon Name Cross-Reference report is shown next.

```
11/12/04              CA Telon NAME CROSS-REFERENCE REPORT          PAGE 1

17:08:40

TLNNAME                                                           QUAL.NAME
 TYPE
-------------------------------------------------------------------------------

HOLD                                                              TELON.HOLD
 JOIN

TRGEMPL                                                           TELON.TRGEMPL
 JOIN


TRGEMPL1                                                          TELON.TRGEMPL1
 TABLE
```

Both reports have three columns that contain the following information:

- A one—to eight-character TLNNAME. The Data Access reports shows this eight-byte name.
- The fully-qualified SQL Table or Join name.
- An indicator of what type the SQL qualified name refers. If the SQL name is the name of an SQL table, this column contains the literal "TABLE." If the SQL name is the name of a JOIN, this column contains the literal "JOIN."

Global Cross-Reference Reports

The CA Telon Global Cross-Reference Reports provide information about one or more specific COBOL or PL/I *data elements*, or character strings, on the TDF VSAM files. The reports also provide information on data elements, or character strings, in PDSes that contain COPY/INCLUDE member, program, and CA Telon source code.

Note: A *data element* is defined in COBOL as a data name, and in PL/I as an identifier.

Some PDS Utility Reports are also produced to audit information about any request for the Global Cross-Reference Report.

The Global Cross-Reference Report and PDS Utility Reports cross-reference data elements or character strings across a set of TDF VSAM files and outside PDSes by:

- Extracting names from specified CA Telon parameters and custom code on TDF VSAM files.
- Concatenating PDSes to the TDF Extract PDS as input to the CA Telon PDS utility (which searches PDSes for specified character strings or COBOL COPY or PL/I INCLUDE members).
- Producing a report that clearly shows all occurrences of data elements or character strings across an application.

Note: You can only search for CA Telon parameters that may be data elements (hvnames). For information on these CA Telon parameters, see "Global Cross-Reference Job Parameters," later in this chapter. The *hvname* is a host variable name, the variable data name declared in the host language (COBOL or PL/I).

Impact analysis

The reports assist you in analyzing the impact of a change to the existing system by:

- Documenting where a data element or character string is used
- Identifying data elements that are no longer referenced

Utilizing this information, you know which modules are impacted when certain variables are changed.

Steps for Producing Global Cross-Reference Reports

The general steps necessary to generate the CA Telon Automated Documentation Global Cross-Reference Reports are outlined in this section.

1. Review the JCL to be used.

   JCL member JUXXRF that invokes PROC TLNUXXRF is provided in the INSTALL data set to produce the Global Cross-Reference reports.

2. Review, and modify if necessary, the parameters. These parameters are described in the next section.

3. Review and update the control cards for input to this PROC.

4. Submit the job and review the results.

Job Parameter Description

This section provides a more detailed description of the parameters used with the Global Cross-Reference Job to give you complete control of the selection and reporting process.

JCL member JUXXRF is provided in the INSTALL data set. This JCL member invokes PROCedure TLNUXXRF.

The parameters specific to this job are described next:

Note: TLNUXXRF parameters TRACE, TLNLIB, SORTLIB, SCRUNIT, and VSQUAL are customized to your installation standards and are part of the CA Telon installation process. These parameters are described in the Installation Guide and also in the "Transport Control File" chapter earlier in this guid.

- **TITLE**—Allows you to customize the title of the resulting report by specifying text to be appended to the report title. Format:

  ```
  TITLE='your title here'
  ```

  The characters between the quotes are appended to the default report title. A dash is used to separate the default title and your customized title.

- **VSQUAL**—VSAM prefix of the TDF VSAM files to be searched.

- **SELCTL**—DSNAME of the data set containing the control card specifying which CA Telon TDF member(s) to be included in the cross-reference report. Format:

  ```
  SELCTL='MY.PDSLIB(PDSMEM)'
  ```

  In this example, the control card is contained in the PDS named MY.PDSLIB and within member PDSMEM.

  The format of the TDF Selection control card is described in the Control Card Descriptions section.

- **SELPARM**—DSNAME of the data set containing additional control cards listing which CA Telon statements are to be searched within the TDF member(s). Format:

  ```
  SELPARM='MY.PDSLIB(PDSMEM1)'
  ```

  In this example, the control cards are contained on the PDS named MY.PDSLIB and within member PDSMEM1.

  The format of these TDF Parameter Selection control cards is described in the Global Cross-Reference Control Cards section.

- **PDS1... PDS9**—DSNAMEs of the PDSes to be included in the global search. All PDSes must have an 80-byte record length, and must contain CA Telon source. You can specify from 0 to 9 PDSes. Format:

  ```
  PDS1='MY.SOURCE.PDS',
  PDS2='MY.COPY.PDS',
  PDS3='MY.COPY2.PDS'
  ```

  In this example, three PDSes are to be included in the search.

  Note: Within the PROC TLNUXXRF, these PDSes are concatenated after a temporary PDS that is created within the job. This temporary PDS has a BLOCKSIZE of 6160.

  If any of the PDS1...PDS9 PDSes has a larger BLOCKSIZE, you must update the PROC TLNUXXRF to make the BLOCKSIZE of the temporary PDS, &.&TDFEXTR., the same as your largest block size.

- **PDSMBR**—Allows you to limit the search of the PDSes to specific members. There are three possible formats:

  ```
  PDSMBR='ALL'
  ```

  The keyword ALL indicates that all members on all PDSes designated in the PDS1...PDS9 parameters are to be included in the search.

  ```
  PDSMBR='aaaaa'
  ```

  Names the specific PDS members to be included in the search. Using this format, only member aaaaa, from any or all of the PDSs designated in the PDS1...PDS9 parameters will be included in the search. The asterisk can be used as a wildcard character in this format. For example, PDSMBR='aa*aa' allows you to include all members with names beginning with aa and ending with aa with any character in the third position, and PDSMBR='aaa*' allows you to include all members beginning with aaa.

  ```
  PDSMBR='TCNTL'
  ```

Points to a data set containing SELECT and EXCLUDE cards which choose some members but exclude others. This is appropriate when a FORMAT 2 mask would not be sophisticated enough to indicate your selection.

The keyword TCNTL indicates that TCNTL is a DDNAME containing PDS member selection control cards. The format of PDS member selection control cards is provided in Global Cross-Reference control cards, later in this section.

To further limit the search, you can use FORMAT 1 and also provide a TCNTL override. This is described in one of the examples later in this section.

- **FUNC**—(*Required.*) Determines which of the three types of global scans to do: a global COBOL search, a global PL/I search, or a search for specific character strings. Format:

  ```
  FUNC=ffffffff
  ```

  where **ffffffff** is one of the following keywords:

  - XREFCBL —The report cross-references a COBOL COPY member. The SRCHCTL parameter, discussed next, gives the name of the COBOL COPY member.
  - XREFPLI —The report cross-references a PL/I INCLUDE member. The SRCHCTL parameter gives the name of the PL/I member.
  - XREFEDIT —The report cross-references a set of character strings. The SRCHCTL parameter provides the DSNAME of a file containing the Search control cards with the character strings.

- **SRCHCTL**—Depending on the value of the FUNC parameter specified above, this parameter provides the name of a COBOL COPY member or a PL/I INCLUDE or the data set name of a file containing a set of character strings. Format:

  ```
  SRCHCTL='MY.PDSNAME(PDSMEM)'
  ```

  - When FUNC=XREFCBL, PDSMEM is a PDS member that must be a COBOL COPY member. CA Telon parses the designated COBOL COPY member to determine all of the data names. CA Telon then searches for and reports on any reference to any of these data names.
  - When FUNC=XREFPLI, PDSMEM is a PDS member which must be a PL/I INCLUDE member. CA Telon parses the designated PL/I INCLUDE member to determine all of the data names. CA Telon then searches for and reports on any reference to any of these data names.
  - When FUNC=XREFEDIT, PDSMEM is a PDS member that contains a set of search control cards. The format of the search control cards is provided in the Global Cross-Reference Control Cards section.

- **SORTCTL**—DSNAME of a data set containing the SORT control cards used to sort the extracted input to the Global Cross-Reference Report. Format:

  ```
  SORTCTL='MY.PDSLIB(PDSMEMB)'
  ```

  In this example, the SORT control cards are contained on the PDS named MY.PDSLIB and within member PDSMEMB.

  Default SORT control cards are included in member CIXSRT provided on the CA Telon INSTALL data set.

  It is not expected that you will need to change this parameter or the default sort sequence, although, as you become more adept at using the Automated Documentation Facility, you can use it as a model for your own processing.

- **XREFOUT**—DSNAME of a temporary data set that contains the information from the TDF and the PDSs before sorting and printing. Format:

  ```
  XREFOUT='MY.PDSLIB(OUTPUT)'
  ```

  In this example, the extracted output is placed in the temporary PDS named MY.PDSLIB and within member OUTPUT.

  It is not expected that you will need to change this parameter, although, as you become more adept at using the Automated Documentation Facility, you can use it as a model for your own processing.

Usage Considerations

This section provides a discussion of some usage details of the Global Cross-Reference process.

Instream Control Card Processing

In some cases, it is preferable to provide the control cards in-stream in the JCL rather than saving them as separate members on a PDS. This is possible by providing DDNAME overrides after invocation of the PROC in the JCL.

The STEPNAMES and DDNAMES used for DDNAME overrides are discussed next.

TDF Selection Control Card

```
Standard Parameter: SELCTL Parameter
STEPNAME  = EXTRACT
DDNAME    = TNREXPIN

DDNAME override:
    //EXTRACT.TNREXPIN DD *
    place your TDF selection control card here
    /*
```

TDF Parameter Selection Control Card

```
Standard Parameter: SELPARM Parameter
NOTE:  These Control Cards are used in two separate steps:
 1st STEPNAME  =  EXTRACT
1st DDNAME    =  TNREXPTB
2nd STEPNAME  =  XREF
2nd DDNAME    =  TNREXPTB

DDNAME overrides:
    //EXTRACT.TNREXPTB DD *
    place your TDF parameter selection control cards here
    /*
    //XREF.TNREXPTB DD *
    repeat your TDF parameter selection control cards here
    /*
```

PDS Member Selection Criteria

```
Standard Parameter: PDSMBR
STEPNAME  =  FIND
DDNAME    =  TCNTL

DDNAME override:
    //FIND.TCNTL DD *
    place your PDS member selection control cards
    here
    /*
```

Search Control Cards

```
Standard Parameter: SRCHCTL
STEPNAME  =  FIND
DDNAME    =  TSYSIN

DDNAME override:
    //FIND.TSYSIN  DD  *
    FIND 'character string' WORD
    FIND 'character string 2' CHARS
    /*
```

For COBOL or PL/I searches (FUNC=XREFCBL or FUNC=XREFPLI) CA Telon internally generates search control cards with the WORD parameter for each of the data names found within either the COBOL COPY or the PL/I INCLUDE member.

The format of the search control card is provided in the Search Control Cards section.

More information:

Search Control Card (see page 133)

Parsing Rules for COPY and %INCLUDE Members

This section describes how CA Telon examines each line of the COBOL COPY or the PL/I %INCLUDE to extract the data names.

For COBOL, the CA Telon PDS Utility checks each line as follows:

```
Get first WORD
If first WORD NUMERIC
   use the next WORD as Data Element (removing trailing periods)
 ELSE
   ignore the card.
```

"INDEXED BY" data elements are not recognized. If you need them, use a separate PDS member with only "INDEXED BY" data elements defined as data items and concatenate it to the COPY member being referenced. For example:

```
05  INDEX-NAME PIC 9.
```

For PL/I, the CA Telon PDS Utility checks each line as follows:

```
If first WORD is 'DCL' or 'DECLARE'
   get the next WORD.
 If current WORD is NUMERIC
   get the next WORD as Data Element
   (removing trailing parentheses, commas, and semi-colons)
 ELSE
   ignore the card.
```

You can have only one data element per line. Only structures are processed. You cannot mix COPY members with "find" cards in the same input.

Sample JCL Submittal

The following pages show you different scenarios for submitting JCL for Global Cross-Reference Reports.

Example 1

The next sample JCL specifies a COBOL Data Element Cross-Reference Report. No "find" cards are specified because CA Telon automatically creates them.

```
//*
//SYSTEM EXEC TLNUXXRF,
//            SELCTL='PDSQUAL.INSTALL(CIESEL)',          1
//            SELPARM='PDSQUAL.INSTALL(CIPSEL)',         2
//            VSQUAL='VSAMQUAL',                         3
//            PDS1='PDSQUAL.PDS1',                       4
//            FUNC=XREFCBL,                              5
//            SRCHCTL='PDSQUAL.PDS(PDSMBR1)',            6
//            PDSMBR='ALL',                              7
//            TITLE='CA Telon TRAINING SYSTEM'           8
//
//
```

1. Member CIESEL on your INSTALL PDS contains the TDF selection control card.
2. Member CIPSEL on the same PDS contains the TDF parameter selection control cards.
3. The high-level qualifier of CA Telon TDF VSAM files is identified in the VSAMQUAL parameter.
4. Only one PDS is to be included in the Global Cross-Reference. This PDS is named pdsqual.PDS1.
5. A COBOL global search is done (XREFCBL).
6. The global search is for all data names found within the COBOL COPY member named PDSMBR1 which is in the PDS named pdsqual.PDS.
7. All members in the PDS to be searched (see item #4 above) are searched for the references.
8. The text indicated within the quotes is appended to the title of the report (in this case, "CA Telon TRAINING SYSTEM").

Example 2

```
//*
//SYSTEM EXEC TLNUXXRF,
//            SELCTL='PDSQUAL.INSTALL(CIESEL)',        1
//            SELPARM='PDSQUAL.INSTALL(CIPSEL)',       2
//            VSQUAL='VSAMQUAL',                       3
//            PDS1='PDSQUAL.TARGET.FILE',              4
//            FUNC=XREFCBL,                            5
//            SRCHCTL='PDSQUAL.PDS(PDSMBR1)',          6
//            PDSMBR='TCNTL',                          7
//            TITLE='CA Telon TRAINING SYSTEM'         8
//FIND.TCNTL DD *                                      9
EXCLUDE MEMBER=PDSMBR7
SELECT MEMBER=PDSMBR*
//
```

1. Member CIESEL on your INSTALL PDS contains the TDF selection control card.
2. Member CIPSEL on the same PDS contains the TDF parameter selection control cards.
3. The high-level qualifier of CA Telon TDF VSAM files is identified in the VSAMQUAL parameter.
4. Only one PDS is to be included in the Global Cross-Reference. This PDS is named pdsqual.TARGET.FILE.
5. A COBOL global search is done (XREFCBL).
6. The global search is for all data names found within the COBOL COPY member named PDSMBR1 which is in the PDS named pdsqual.PDS.
7. The DDNAME TCNTL contains a set of PDS selection control cards.
8. The text indicated within the quotes is appended to the title of the report (in this case, CA Telon TRAINING SYSTEM).
9. The DDNAME TCNTL is provided as a DDNAME override to the PROCSTEP FIND.

Example 3

```
//*
//SYSTEM EXEC TLNUXXRF,
//            SELCTL='PDSQUAL.INSTALL(CIESEL)',          1
//            SELPARM='PDSQUAL.INSTALL(CIPSEL)',         2
//            VSQUAL='VSAMQUAL',                         3
//            PDS1='PDSQUAL.PDS1',                       4
//            FUNC=XREFEDIT,                             5
//            PDSMBR='TR******',                         6
//            TITLE='CA Telon TRAINING SYSTEM'           7
//FIND.TSYSIN DD *                                       8
FIND EMPL-DOB WORD
FIND EMPL-ID CHARS           <---Supply multiple "find"
FIND EMPL-NAME SUFFIX             cards instead of a COPY
//                                or INCLUDE member.
```

1. Member CIESEL on a PDS named pdsqual.INSTALL contains the TDF selection control card.
2. Member CIPSEL on the same PDS contains the TDF parameter selection control cards.
3. The high-level qualifier of CA Telon TDF VSAM files is identified in the VSAMQUAL parameter.
4. Only one PDS is to be included in the Global Cross-Reference. This PDS is named pdsqual.PDS1.
5. A global search is done. (XREFEDIT).
6. Only members in the PDS to be searched (see item 4 above) that begin with the characters TR are to be searched for the references.
7. The text indicated within the quotes is appended to the title of the report (in this case, CA Telon TRAINING SYSTEM).
8. The global search is done for the character strings given in the search control cards given in the TSYSIN data set. This example shows instream DDNAME override instead of using the SRCHCTL parameter.

Example 4

The next sample JCL specifies how to exclude PDS members that you do not want to search:

```
//*
//SYSTEM EXEC TLNUXXRF,
//            SELCTL='PDSQUAL.INSTALL(CIESEL)',       1
//            SELPARM='PDSQUAL.INSTALL(CIPSEL)',      2
//            VSQUAL='VSAMQUAL',                      3
//            PDS1='PDSQUAL.TARGET.FILE',             4
//            FUNC=XREFCBL,                           5
//            SRCHCTL='PDSQUAL.PDS(PDSMBR1)',         6
//            PDSMBR='ALL',                    <---  7
//            TITLE='CA Telon TRAINING SYSTEM'        8

//FIND.TCNTL DD *                                     9
EXCLUDE MEMBER=PDSMBR7       <---This searches all
//                                 members except PDSMBR7.
```

1. Member CIESEL on your INSTALL PDS contains the TDF selection control card.
2. Member CIPSEL on the same PDS contains the TDF parameter selection control cardss.
3. The high-level qualifier of CA Telon TDF VSAM files is identified in the VSAMQUAL parameter.
4. Only one PDS is to be included in the Global Cross-Reference. This PDS is named pdsqual.TARGET.FILE.
5. A COBOL global search is done (XREFCBL).
6. The global search is for all data names found within the COBOL COPY member named PDSMBR1 which is in the PDS named pdsqual.PDS.
7. All members in the PDS to be searched (see item 4) are searched (see also item 9 below).
8. The text as indicated is concatenated to the report title.
9. This is an in-stream DDNAME override to provide PDS member selection control cards. While item 7 says ALL PDS members are included in the search, this DDNAME override to TCNTL refines that criteria.

   In fact, all PDS members except member PDSMBR7 are searched.

   Note: In this example, the same result would have been provided if PDSMBR='TCNTL' had been used.

JCL DDNames

The following statement descriptions identify all DDnames identified in the detail Automated Documentation procedure TLNUXXRF:

- **NULLPDS1**
- **NULLPDS2**
- **NULLPDS3**
- **NULLPDS4** Null PDSes act as place holders for concatenation.
- **NULLPDS5**
- **NULLPDS6**
- **NULLPDS7**
- **NULLPDS8**
- **NULLPDS9**
- **STEPLIB**—Library where the CA Telon load modules are located.
- **TNSYSLIB**—Library where the CA Telon load modules are located.
- **TLTRACE**—Used for CA Telon database call-tracing. Used only if the TRACE parameter = Y. This is used in debugging problems.
- **SYSOUT**
- **SYSPRINT**—System output files used for debugging purposes.
- **SYSUDUMP**:
- **SORTWK01**
- **SORTWK02**
- **SORTWK03**—Work files used while processing sorts.
- **SORTWK04**
- **SORTWK05**
- **TNTCCL**—TDF VSAM Custom Code File.
- **TNTDD**—TDF VSAM Data Administration File.
- **TNTDDW**—TDF VSAM Data Administration Work File.
- **TNTDF**—TDF VSAM Entity File.
- **TNTDFW**—TDF VSAM Entity Work File.
- **TNTDX**—TDF VSAM Data Administration File for SQL information.
- **TNREXPIN**—File to contain TDF selection control cards.
- **TNREXPTB**—File to contain TDF parameter selection control cards.
- **TNADREXP**—PDS to contain output from TDF extract. Must be a new temporary PDS.

- **TCNTL**—Input Control File for PDS member selection control cards.
- **TPDS1**—Members are selected from these PDS libraries to be processed. The data sets are identified using PDS1, PDS2, PDS3, and so on.
- **TSYSPRT**—Utility SYSPRINT data set containing the input control card report and any error messages.
- **TPRINT**—Utility SYSPRINT file containing the Members Processed and Search Element Summary reports.
- **TSYSOUT**—Extracted cross-reference data file that is used to create the final Cross-Reference Report.
- **TSYSIN**—Contains the search control cards.
- **SORTLIB**—System sort load library.
- **SORTIN**—Extracted cross-reference data that used to create the final Cross-Reference Report.
- **SORTOUT**—Sorted Cross-Reference that is used to create the final Cross-Reference Report.
- **SYSIN**—Control cards to sort the extracted cross-reference information.
- **TNINFILE**—Sorted, cross-reference information—ready to print.
- **TNRXLGD**—Legend of Cross-Reference Report.
- **TNRXREF**—Body of the Cross-Reference Report.
- **TNRXIDX**—Index of the Cross-Reference Report.

Control Card Descriptions

There are four kinds of control cards involved:

- TDF Selection
- TDF Parameter Selection
- PDS Member Selection
- Search

TDF Selection Control Card

Usually kept as a member of a PDS, this control card is referenced in the JCL/PROC parameter SELCTL.

A default control card is provided in member CIESEL on the INSTALL data set.

Format

```
        1   1   2   2   3
....5....0....5....0....5....0....
 NNNNHI,T1,T2,T3,T4,T5
```

Field descriptions

- **Parameter**—NNNN
  - **Field**—Global Cross-Reference
  - **Length**—4
  - **Req/Opt**—Required
  - **Description**—Required value is XREF.
- **Parameter**—H
  - **Field**—Entity Header
  - **Length**—1-5
  - **Req/Opt**—Required
  - **Description**—You must enter either a specific header or the keyword NONE. If you enter NONE, the TDF is not included in the global search, and any other parameters on this control card are ignored. If you provide a specific header, all TDF entities that have an entity header equal to it are selected.

- **Parameter**—I
  - **Field**—ID
  - **Length**—1-5
  - **Req/Opt**—Optional
  - **Description**
    - This is the CA Telon entity ID. It is entered on the control card concatenated to the Header (when you have given a specific header).
    - While you had to enter a specific header to be searched, you can enter either a specific or a generic ID by using a wildcard.
    - The asterisk (*) can be used as a trailing wildcard selector. If you want a range of IDs, enter a partial ID followed by asterisks. Once the program encounters an asterisk (a wildcard) it assumes asterisks for the remainder of the ID.
- **Parameter**—T1,T2,T3,T4,T5
  - **Field**—Entity Types
  - **Req/Opt**—Optional
  - **Description**—Values are:
    - **ALL**—All TYPEs are searched
    - **NONE**—No TYPEs are searched
    - **PD**—Panel Definition
    - **PI**—Panel Image
    - **SD**—Screen Definition
    - **BD**—Batch Definition
    - **DR**—Driver Definition
    - **RD**—Report Definition
    - **ND**—CICS Non-Terminal Definition
    - **SP**—Stored Procedure Definition

- ENTITY types are separated from each other and from the HEADER-ID field by commas.
- If you have specified that the TDF will be searched, by providing a specific Entity Header and an ID, you must provide at least one ENTITY type.
- If you have specified that the TDF will not be searched, by specifying NONE in the HEADER field, no ENTITY types need be provided.
- You are allowed to provide up to five ENTITY types on one control card.
- All TDF entities that have an entity type equal to the value of the given field(s) are selected.

Sample control cards are shown next:

Example 1

```
        1    1    2    2    3
----5----0----5----0----5----0
XREFTRMENU,PD,RD,BD
```

Example 1 requests only TDF entities of PD, RD, and BD with header-ID = TRMENU.

Note: No SD is selected.

Example 2

```
        1    1    2    2    3
----5----0----5----0----5----0
XREFTR****,SD
```

Example 2 requests all SDs for HEADER TR and any ID.

Example 3

```
      1   1   2   2   3
----5----0----5----0----5----0
XREFTRMENU,ALL
```

Example 3 requests all Types for the specific HEADER and ID of TRMENU.

Example 4

```
        1    1    2    2    3
----5----0----5----0----5----0
XREFTR****,PD,BD
```

Example 4 requests all BDs and all PDs for Header-ID starting with TR.

Note: No SDs, DRs, SPs or RDs are selected.

Example 5

```
        1    1    2    2    3
----5----0----5----0----5----0
XREF,NONE
```

Example 5 requests not to search the TDF.

Sample JCL

The next sample JCL illustrates how to specify SELCTL control cards:

```
//*
//SYSTEM EXEC TLNUXXRF,
//           SELCTL='PDSQUAL.INSTALL(CIESEL)'   <---
             .
             .
             .
//
```

where PDSQUAL.INSTALL(CIESEL) contains the control card specifying the entities that you want to search for in the TDF VSAM files.

If you do not want to use the default data set, you can specify your own data set with SELCTL. For example:

SELCTL='your.data.set'.

TDF Parameter Selection Control Card

This control card set is referenced in the JCL/PROC parameter SELPARM. A default control card set is provided in member CIPSEL on the INSTALL data set.

Format

```
        1    1    2    2    3
....5....0....5....0....5....0....
 CCCCCCCC
```

Field descriptions

- **Parameter**—CCCCCCCC
  - **Field**—CA Telon Parameter Name
  - **Columns**—1-8
  - **Req/Opt**—Optional
  - **Description**
    - One CA Telon parameter should be entered on any single card. Multiple TDF parameter selection control cards can be provided.
    - control cards can be provided in any order.
    - Mistyped or invalid control cards will be bypassed and ignored.
    - If you do not specify any TDF parameter selection control cards, the default is for all TDF parameters to be searched.
    - If the TDF Selection control card indicates that the TDF is included in the global search, these TDF parameter selection control cards are used to further define the limits of the TDF search.

Sample TDF parameter selection control cards are shown next:

```
CALC
DBNAME
FDREC
WHEN
```

The following list shows CA Telon parameters which can be input in this control card set. The first column is the list of values to be used. The second column shows the CA Telon Statements with which the various parameters are associated.

Parameter—Statements

- **SRC** —SRC
- **A4EMSG**—IMSDRV, IMSPGM
- **ALIAS**—DCLCOL
- **CALC**—FIELD
- **COBVSKY**—SEGMENT
- **COND**—XFEDIT
- **COPYLBL**—SEGMENT, TABLE
- **CTLPIC**—GROUP
- **CTLVAR**—GROUP
- **DBNAME**—FIELD
- **DCLHOST**—DCLCOL
- **DCLLBL**—TABLE
- **ERRMSG**—SEGEDIT, XFEDIT
- **FDREC**—CREATE, UPDATE
- **GENKEYL**—DELETE, READ, SEGEDIT, SEGMENT, UPDATE
- **HAVING**—BROWSE, READNEXT, OUTREAD, OIREAD, INREAD, TRANSACT, CREATE, DELETE, READ, UPDATE
- **IEXTEND**—FIELD
- **IOAREA**—BROWSE, READNEXT, OUTREAD, OIREAD, INREAD, TRANSACT, CREATE, DELETE, READ, UPDATE, SEGEDIT, DLIDSC
- **ISEGIDX**—SEGLOOP
- **KEY**—DCLCOL, ROW, DLIDSC, BROWSE, READNEXT, OUTREAD, OIREAD, INREAD, TRANSACT, CREATE, DELETE, READ, UPDATE
- **KEYFEED**—DLIDSC
- **MAPOUT**—FIELD
- **OEXTEND**—FIELD
- **OF**—FIELD

- **OSEGIDX**—SEGLOOP
- **PAGEKEY**—SEGLOOP
- **PRINT**—GROUP
- **SAVEKEY**—SEGLOOP
- **SCHFLDC**—SEGLOOP
- **SEGLTH**—CREATE, READ, SEGEDIT, UPDATE, SEGMENT
- **SELKEY**—FIELD
- **SSALIST**—CREATE, DELETE, READ, SEGEDIT, UPDATE
- **STBRKEY**—SEGLOOP
- **TPPARMS**—CREATE, READ
- **WHEN**—SEGEDIT
- **WHERE**—ROW, BROWSE, READNEXT, OUTREAD, OIREAD, INREAD, TRANSACT, CREATE, DELETE, READ, UPDATE

PDS Member Selection Control Card

This control card set is specified as an override to the DDname TCNTL in Step FIND, when the PROC parameter PDSMBR='TCNTL'. See Example 2, presented in Global Cross-Reference Reports, earlier in this chapter.

Format

```
        1   1   2   2   3
....5....0....5....0....5....0....
 aaaaaaa MEMBER=nnnnnnnn
```

Field descriptions

- **Parameter**—aaaaaaa
  - **Field**—Action for PDS MEMBER
  - **Length**—7
  - **Req/Opt**—Required
  - **Description**—Values are:

    SELECT —Indicates that any member with the given name on any of the PDSes specified should be included in the cross-reference report

    EXCLUDE —Indicates that any member with the given name on any of the PDSes specified should not be included in the cross-reference report

- **Parameter**—MEMBER=
  - **Field**—Literal
  - **Description**—Should be separated from the ACTION by at least one space.
- **Parameter**—nnnnnnnn
  - **Field**—PDS MEMBER Name
  - **Length**—1-8
  - **Req/Opt**—Required
  - **Description**

    Should immediately follow the MEMBER= literal.

    This must be a valid PDS member name. The asterisk can be used as a trailing wildcard character.

    Multiple PDS selection control cards can be specified, in any order.

    Mistyped or invalid PDS selection control cards are ignored.

Sample Control PDS Member Selection Cards are shown next:

Example 1

```
        1   1   2
----5----0----5----0---
EXCLUDE MEMBER=PDSMBR7
```

Searches all members except PDSMBR7.

Example 2

```
        1   1   2
----5----0----5----0---
EXCLUDE MEMBER=PDSMBR7
SELECT MEMBER=PDSMBR*
```

Searches all members PDSMBRA through PDSMBR9 except PDSMBR7.

Example 3

```
        1   1   2
----5----0----5----0---
EXCLUDE MEMBER=GLS*
```

Searches all members, except anything beginning with the characters GLS.

Example 4

```
        1   1   2
----5----0----5----0---
EXCLUDE MEMBER=PDSMBR7
EXCLUDE MEMBER=GLS444
SELECT MEMBER=PDSMBR*
SELECT MEMBER=GLS*
```

Includes all members PDSMBR1 through PDSMBR9 except PDSMBR7 and also includes any member beginning with GLS except GLS444.

Search Control Card

This control card set is referenced in the JCL/PROC parameter SRCHCTL.

Format

```
         1    1    2    2    3
....5....0....5....0....5....0....
FIND character-string,parameter
          or
FIND parameter,character-string
```

Field descriptions

- **FIND**—Literal
- **Length**—4
- **character-string**—Character string to be searched for in the global search. The character string should be surrounded by single quotes (apostrophes) if it contains any special characters or blanks.

  A TEXT STRING is a special type of a character string. If you want the contents of the character string to be treated as if it were all in uppercase, proceed and follow the quotes with the letter T.

  The character string can be separated from the parameter by either a comma or a space.

  The character string can either proceed or follow the parameter.
- **parameter**—One of four keyword parameters, telling the search how to treat the character string. This is an optional parameter, and need not be coded on the control card. Values are:
  - CHARS (default)—The search reports on all occurrences of the character string, wherever it occurs, whether alone, or imbedded within another data name or label. When no parameter is coded, the search treats the character string as a CHARS.
  - WORD—The search reports on occurrences of the character string only if it occurs as an independent data name or label. The search does not report any data names or labels that may have this character string imbedded within.
  - SUFFIX—The search treats the character string as a suffix and reports only data names or labels where this suffix occurs.

    CA Telon searches for occurrences of the character string that are preceded by an alphameric character and followed by a non-alphameric character.
  - PREFIX—The search treats the character string as a prefix and reports only data names or labels where this prefix occurs.

CA Telon searches for occurrences of the character string that are preceded by a non-alphameric character and followed by an alphameric character.

Multiple Search control cards can be specified, in any order.

Mistyped or invalid SEARCH control cards will appear on an error report, but are ignored in the global search.

Sample search control cards are as follows:

Example 1

```
        1   1   2
----5----0----5----0---
FIND EMPL
```

Searches for and reports on all occurrences of the character string EMPL.

Example 2

```
        1   1   2
----5----0----5----0---
FIND 'EMPL'
```

Searches for and reports on all occurrences of the character string EMPL.

```
        1   1   2
----5----0----5----0---
FIND 'EMPL',CHARS
```

Same as above, since CHARS is the default parameter.

Example 3

```
        1   1   2
----5----0----5----0---
FIND PREFIX,'EMPL'
```

Searches for and reports on all data names or labels where the character string EMPL is a PREFIX.

Example 4

```
        1   1   2
----5----0----5----0---
FIND CTL SUFFIX
```

Searches for and reports on all data names or labels where the character string CTL is a SUFFIX.

Example 5

```
        1   1   2
----5----0----5----0---
FIND WORD 'EMPL-NAME'
FIND 'TO EMPLOYEE-TOTAL'
```

Searches for and reports any reference to the data name or label EMPL-NAME. Also searches for and reports any reference to the character string TO EMPLOYEE-TOTAL.

Example 6

```
        1   1   2
----5----0----5----0---
FIND T'to employee-total'T
```

Searches for and reports on any occurrence of the character string TO EMPLOYEE-TOTAL.

Examples—Global Cross-Reference Report

The Global Cross-Reference Report provides information about one or more specific data elements or character strings. The report is a language-specific report (COBOL or PL/I) that can be tailored to search all or a portion of a specified entity or entities in TDF VSAM files. You can also specify up to nine PDSes containing COPY/%INCLUDE member, program, and CA Telon source code.

The Global Cross-Reference Report consists of three parts:

- The **Legend** page lists the CA Telon parameters and parameters that you selected. It also shows the file prefix of the VSAM files used to create the extract, and the data set names of the PDS or PDSes selected. Finally, it provides a guide to the formats of the Reference Information section of the main report.
- **Multiple data elements, or character strings** are shown per page, alphabetically. Header and footer 'tags' at the extreme right assist you in scanning the report.
- The **Index** shows the data elements, or character strings (in alphabetic order for the COPY member or "find" cards provided), the number of source code lines for each search control card item found, and points to the first page number on the report where the item can be found.

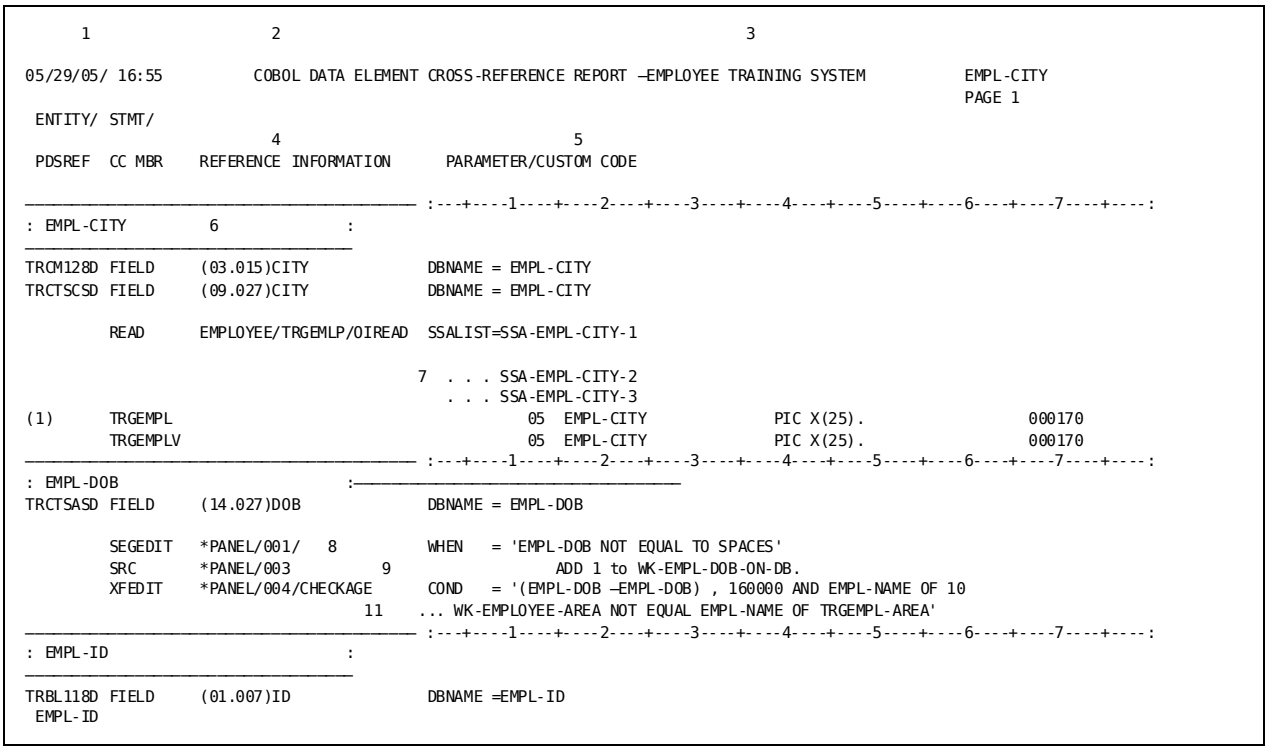
The following pages contain examples of the three sections of the Global Cross-Reference Report. The explanations for each field follow each example.

```
      1                                2

05/29/05 16:53                      LEGEND OF COBOL DATA ELEMENT CROSS-REFERENCE REPORT —COPY MEMBER CHANGE ANALYSIS

   LIBRARIES ACCESSED:


    3    TDF VSQUAL:  CHSO1VS.DEVTST
         PDS (1) —CHSO2.REL14.SOURCEC     4
         PDS (2) —CHSO1.CURRENT.SOURCE
         PDS (3) —CHSO1.TEST<SOURCE

THE FOLLOWING LIST OF CA Telon PARAMETERS AVAILABLE FOR THE ELEMENT CROSS-REFERENCE REPORT HAS BEEN MARKED TO SHOW WHICH
PARAMETERS WERE SELECTED FOR THIS PARTICULAR REPORT.  A GUIDE TO THE FORMAT FOR THE REFERENCE INFO HAS ALSO BEEN
PROVIDED WHICH SHOWS THE STATEMENT TO WHICH IT APPLIES.

*-------------------------------------------------------------------------------------------------------------------

 *   PARAMETER  SEL  FORMAT(S)      PARAMETER SEL FORMAT(s)           PARAMETER SEL FORMAT(S)    PARAMETER SEL FORMAT(S)

 *    A4EMSG     N      D            ERRMSG   N     E                OF        Y     B           SELKEY    Y      B
 *    CALC   5   Y      B            FDREC    Y     A                OSEGIOX   Y   D           SSALIST   Y       A,E
 *    COBVSKY    Y  6   F            GENKEYL  Y   A,E,F   7       PAGEKEY   Y     D
       STBRKEY   Y      D
 *    COND       Y      E            IEXTEND  N     B                PRINT     Y     C           TPPARMS   N      A
 *    COPYLBL    N      F            IOAREA   T   A,E,F              SAVEKEY   Y     D           WHEN      Y      E
 *    CTLPIC     N      C            ISEGIDX  Y     D                SCHFLDC   N     D

 *    CTLVAR     Y      C            MAPOUT   Y     B                SEGKEY    Y   A,E,F

 *    DBNAME     Y      B            DEXTEND  N     B                SEGLTH    Y   A,E,F


                8  ...   EXPLANATION OF FORMAT(s) COLUMN VALUES ...

 FORMAT A:  CREATE, DELETE, READ, AND UPDATE STATEMENTS
      PCBNAME/SEGMENT LABEL(OR DBSEG, IF NONE)USER-I/O) LABEL(REQUEST, IF NONE)

 FORMAT B;  FIELD STATEMENT
      (POSITION)FIELD NAME (IF NAMED FIELD, ONLY POSITION IF UNNAMED)


 FORMAT C:  GROUP STATEMENT
      GROUP NAME/USAGE

 FORMAT D:  IMSORV, IMSPGM, AND SEGLOOP STATEMENTS DO NOT REQUIRE REFERENCE INFORMATION

 FORMAT E:  SEGEDIT, SRC, AND XFEDIT STATEMENTS
      PANEL OR (POSITION)SELECT FIELD NAME/SEQUENCE NUMBER/SEGEDIT NAME OR XFEDIT NAME(EXCEPT SRC STATEMENT)

 FORMAT F:  SEGMENT STATEMENT
      PCBNAME/SEGMENT LABEL(DBSEG, IF NO LABEL)
```

The numbers in the following description refer to the preceding report.

Numbers	Description
1	Specifies the date and time of the execution of the report.
2	Specifies the name of the function that you requested by using the FUNC parameter of TLNUXXRF. This appears on COBOL, PL/I, or EDIT.

Numbers	Description
3	Specifies the file prefix of the TDF VSAM files used to create the extract. You supplied this file prefix on the VSQUAL parameter of TLNUXXRF.
4	Specifies the data set name of the PDS that you supplied as PDS1, PDS2, and so on. The number in parentheses helps you match up the PDSREF number on the body of the report so that you can determine which PDS contains the data elements for which you are searching.
5	Shows the status whether it was selected for inclusion in this report or not, for each CA Telon parameter.
6	Lists the CA Telon parameters available for this report. This list contains four columns and shows all of the CA Telon parameters that are available.
7	Shows the format identifier for each CA Telon parameter. Use the format identifier letter to refer the explanation of format columns given. **Note:** Reference information is information given to aid in identifying where on the TDF the data element exists; it applies at the statement level.
8	Shows all the possible formats available for the Reference Information on the body of the report for this specific CA Telon parameter. For example, the SSALIST parameter can be found on either the statements specified by FORMAT A or FORMAT E of field 8. These formats help you interpret the reference information on the Global Cross-Reference Report page.



```
      1                  2                                                3

05/29/05/ 16:55          COBOL DATA ELEMENT CROSS-REFERENCE REPORT –EMPLOYEE TRAINING SYSTEM          EMPL-CITY
                                                                                                     PAGE 1
 ENTITY/ STMT/
                          4                                  5
 PDSREF  CC MBR    REFERENCE INFORMATION      PARAMETER/CUSTOM CODE

————————————————————————————————————— :---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----:
: EMPL-CITY         6             :
————————————————————————————————————
TRCM128D FIELD     (03.015)CITY             DBNAME = EMPL-CITY
TRCTSCSD FIELD     (09.027)CITY             DBNAME = EMPL-CITY

         READ      EMPLOYEE/TRGEMLP/OIREAD  SSALIST=SSA-EMPL-CITY-1

                                          7  . . . SSA-EMPL-CITY-2
                                             . . . SSA-EMPL-CITY-3
(1)      TRGEMPL                                    05  EMPL-CITY              PIC X(25).                  000170
         TRGEMPLV                                   05  EMPL-CITY              PIC X(25).                  000170
————————————————————————————————————— :---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----:
: EMPL-DOB                          :————————————————————————————————
TRCTSASD FIELD     (14.027)DOB              DBNAME = EMPL-DOB

         SEGEDIT   *PANEL/001/   8          WHEN   = 'EMPL-DOB NOT EQUAL TO SPACES'
         SRC       *PANEL/003          9                ADD 1 to WK-EMPL-DOB-ON-DB.
         XFEDIT    *PANEL/004/CHECKAGE      COND   = '(EMPL-DOB –EMPL-DOB) , 160000 AND EMPL-NAME OF 10
                                  11     ... WK-EMPLOYEE-AREA NOT EQUAL EMPL-NAME OF TRGEMPL-AREA'
————————————————————————————————————— :---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----:
: EMPL-ID                           :
————————————————————————————————————
TRBL118D FIELD     (01.007)ID               DBNAME =EMPL-ID
 EMPL-ID
```

The numbers in the following description refer to the preceding report.

Numbers	Description
1	Specifies the date and time of the execution of the report.
2	Specifies the name of the function that you requested by using the FUNC parameter of TLNUXXRF. This appears on COBOL, PL/I or EDIT.
3	Specifies the user title of the report. You supply this title in the TITLE parameter.
4	Provides the reference information. The format depends on the type of parameter. For example, the CA Telon DBNAME parameter REFERENCE INFORMATION has the format: (POSITION)FIELD NAME The REFERENCE INFORMATION field has many different formats. For more information, see Explanation of Legend Page. Custom code members and COPY/%INCLUDE members do not have Reference Information.
5	Shows the CA Telon parameter and value, of either custom code or COPY member source code (the actual line of code being searched for).

Numbers	Description
6	Shows the data element that was found in the source libraries specified. This field is sequenced alphanumerically in ascending order.
7	Shows the list parameters. Three dots indicate multiple hvname values for the parameter. In this case the SSALIST parameter has three values.
8	Indicates that source statements do not go beyond column 72 to avoid confusion with line numbers.
9 through 11	Indicates that character strings are not split if they extend past column 72. Instead, the entire string is moved to the next line. The three dots here indicate continuation.

```
05/29/05 16:56          COBOL DATA ELEMENT CROSS-REFERENCE REPORT -EMPLOYEE TRAINING SYSTEM            EMPL-ID
                                                                                                       PAGE 2
 ENTITY/ STMT/

 PDSREF  CC MBR    REFERENCE INFORMATION      PARAMETER/CUSTOM CODE
-------------------------------------- :---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----:
: EMPL-CITY                      : CONTINUED  2
--------------------------------
TRBL12BD FIELD     (01.007)ID              DBNAME =EMPL-ID
    3    CCMEM1                      4                MOVE EMPL-ID TO EX-EMPL-ID
                  001060
TRBL41BD FIELD     (01.004)       5        DBNAME =EMPL-ID
TRBL42BD FIELD  6  (01.004)IDH1            DBNAME =EMPL-ID
         FIELD     (01.004)ID              DBNAME =EMPL-ID
TRCM12BD FIELD     (01.006)ID              DBNAME =EMPL-ID
TRCM41BD FIELD     (01.007)ID              DBNAME =EMPL-ID
         GROUP     ASTERISK/CONTROL   7    CTLVAR =EMPL-ID
TRCTSASD UPDATE    EMPLOYEE/TRGEMPL/OUTREAD SEGKEY =EMPL-ID
         FIELD     (04.040)ID              DBNAME =EMPL-ID       .
TRCTSLSD SEGLOOP                           PAGEKEY=EMPL-ID
         SEGLOOP                           SAVEKEY=EMPL-ID       .
                                      8     . . . XFER-EMPL-ID
         FIELD     (08.007)ID              DBNAME =EMPL-ID
    9
(1)      TRGEMPL                                    10  EMPL-ID          PIC X(6)                        000300
    10   TRGEMPLV   11                              10  EMPL-ID          PIC X(6)                            000300
-------------------------------------- :---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----:
: EMPL-NAME                      :
--------------------------------
TRCM12BD FIELD     (01.015)NAME            DBNAME =EMPL-NAME
TRCTSTSD FIELD     (04.056)NAME            DBNAME =EMPL-NAME OF WK-EMPLOYEE-AREA
                                  12             OF WORK-SEGMENT-IN
                                           DBNAME =EMPL-NAME OF TRGEMPL-AREA                          000400
(1)      TRGEMPL                                    05  EMPL-NAME        PIC X(25).                      000400
         TRGEMPLV                                   05  EMPL-NAME        PIC X(25)

  13
EMPL-NAME
```

The numbers in the following description refer to the preceding report.

Numbers	Description
1	The first data element on this page.

Numbers	Description
2	Indicates that this page is a continuation page. The previous page contains additional information on this data element.
3	Identifies a TDF entity name requested by the SELCTL parameter of TLNUXXRF. Spaces indicate the TDF entity name above it.
4	This line is an example of a custom code member. It belongs to the TDF name on the same line.
5	An example of position (line and column). The line directly underneath (and most FIELD statements) also has a CA Telon field name. The CA Telon field name is defined in the PD.
6	The TELON statement or custom code member name.
7	This line is an example of a CA Telon parameter which only applies to CA Telon Batch Programs.
8	The three dots indicate continuation of the above parameter.
9	This is the line number. CA Telon generates this to help you find the line of custom code more easily.
10	The PDSREF of (1) means that these members are both located in the PDS referenced on the LEGEND page as PDS1 and supplied to the JCL with the PDS1 parameter.
11	This is an example of a COPY member with no Reference Information.
12	The OF parameter automatically appears with the DBNAME parameter (even if not specifically requested), and the DBNAME parameter automatically appears with the OF parameter. This assists you in identifying the proper field. The OF literal precedes the OFNAME. A second DBNAME has been mapped for this field.
13	The last data element on this page.

```
      1                          2                                                 3
05/29/05 16:56          INDEX OF COBOL DATA ELEMENT CROSS-REFERENCE REPORT –TEST SYSTEM ONLY
 PAGE  COUNT  DATA ELEMENT                 SEARCH CONTROL INPUT                                                    PAGE    I-1

             ——————————————————————————— :---+----1----+----2----+---3----+----4----+---5----+----6----+---7----+----:
  4      5             6                      8
  1     13   EMPL-CITY                                 05   EMPL-CITY            PIC X(25).

         0   EMPL-DEPARTMENT                           05   EMPL-DEPARTMENT      PIC XXX.
  1      8   EMPL-DOB                                  05   EMPL-DOB             PIC 9(6).
         0   EMPL-DOE                                  05   EMPL-DOE             PIC 9(6).
         0   EMPL-HOURLY-RATE                          05   EMPL-HOURLY-RATE     PIC 999V99 COMP-3.

         0   EMPL-HOURS-PER-WEEK                       05   EMPL-HOURS-PER-WEEK  PIC 999V9
  2     45   EMPL-ID                                        10  EMPL-ID          PIC X(6).

         0   EMPL-KEY                                  05   EMPL-KEY
  2     17   EMPL-NAME                                 05   EMPL-NAME            PIC X(25).
         0   EMPL-PHONE                                05   EMPL-PHONE           PIC X(10).
         0   EMPL-SEX                                  05   EMPL-SEX             PIC X.
         0   EMPL-STATE                                05   EMPL-STATE           PIC XX.
```

```
05/29/02 16:56          INDEX OF COBOL DATA ELEMENT CROSS-REFERENCE REPORT –PRODUCTION SYSTEM
 PAGE  COUNT  DATA ELEMENT                 SEARCH CONTROL INPUT                                                    PAGE    I-1


             ——————————————————————————— :---+----1----+----2----+---3----+----4----+---5----+----6----+---7----+----:
  1     13   EMPL-CITY                 FIND EMPL-CITY
  1     13   EMPL-DEPARTMENT           FIND EMPL-DEPARTMENT
  1      8   EMPL-DOB                  FIND EMPL-DOB
```

The numbers in the following description refer to the preceding report.

Numbers	Description
1	The date and time of the execution of the report.
2	The function that you requested by using the FUNC parameter. This will appear as COBOL, PL/I, or EDIT.
3	The user title of the report. You supplied the user title using the TITLE.
4	The page of this report which contains the first occurrence of the data element shown on this line. If the data element was not found, the page number is spaces and count is zero.
5	The number of times that the data element was found in the libraries requested. The value is zero if the data element was not found.
6	This field contains the data element name which was used to search the supplied libraries. It is sequenced alphanumerically in ascending order.

Numbers	Description
7	A blank line appears after every five data elements. This improves the readability of the report.
8	Search Control Input is the TSYSIN input card image. The first sample shows a COBOL COPY member as input.

Examples—CA Telon PDS Utility Reports

The CA Telon PDS Utility Reports are audit reports that are generated to provide statistics and audit information about the request for the Global Cross-Reference Report.

The following CA Telon PDS Utility reports are provided:

- **Input Control Cards**—Shows the cards used to specify the scope of the search (and error messages if the cards were not properly specified).
- **Utility Members Processed**—Shows the DSNAME and MEMBER names and the number of times the data elements were found.
- **Search Element Summary**—Shows the search control cards (either COPY member, %INCLUDE member, or "find" cards, in the sequence they exist in the input data set), and the number of times each individual search control card item was found.

Input Control Cards Report

The following pages contain examples of three CA Telon PDS Utility reports. The explanations for each field follow the example.

Sample report

A sample of the input control cards report is shown next.

```
05/29/05          CA Telon PDS UTILITY –INPUT CONTROL CARDS          PAGE 01
   1                                                    2    XREFCBL
  DDNAME:TCNTL     –MEMBER SELECTION CARDS
  ___________________________________
3 SELECT MEMBER=TRCM**SD    4
  EXCLUDE MEMBER=TRCM12SD

 DDNAME:TSYSIN –CONTROL CARDS
 ____________________________________
 FIND XFER-TASK

 FIND EMPL-KEY     WORD

 FIND TASK         PREFIX          5

 FIND TIME-INDEX   WORD

 FIND ID           SUFFIX
 FIND 'INDEXED-BY'
```

Detail information

The following numbers correspond to the numbers on the preceding report.

Number	Description
1	Date of the execution of the report.
2	Function that you requested by using the FUNC parameter.
3	A list of all SELECT and/or EXCLUDE cards provided.
4	Member identification field. This contains either a complete member name or a member name with a wildcard position. You entered the member identification field on the SELECT and/or EXCLUDE cards.
5	Control cards you supplied to specify which data elements to "find." This section of the report is only produced if you specify FUNC=XREFEDIT.

Utility Members Processed Report

Sample report

A sample of the Utility Members Processed report is shown next.

```
05/29/02         CA Telon PDS UTILITY –MEMBERS PROCESSED              PAGE 01
   1                                                    2     XREFCBL
   3          4            5
   MEMBER     COUNT             DATA SET NAME
   ______     _____        ________________________
   EXTRACT    0013             TELON.TEST.SOURCLIB
   HLDDBDV1   0000

   HLPDBDV1   0000             &.&TDFEXTR.
   TRBPSORT   0003
   TRCM12BD   0010             TELON.PRODUCTION SOURCLIB
   TRCM20BD   0009             &.&TDFEXTR.
   TRCM41BD   0014             TELON.TEST.SOURCLIB
```

Detail Information

The following numbers correspond to the numbers on the preceding report.

Number	Indicates
1	Date of the execution of the report.
2	Function that you requested by using the FUNC parameter.
3	List of PDS members that were searched. These members are determined by the PDSMBR parameter.
4	Match count that identifies the number of times a data element was found in this member.
5	Name of the data set upon which the member was found. A value of &.&TDFEXTR. indicates that the member was found on the CA Telon TDF and was extracted to the temporary PDS for the duration of the scan. A blank line indicates the data set name is the same as the one on the line immediately above.

Search Element Summary Report

Sample report

A sample of the Search Element Summary report is shown next.

```
1
 05/29/05       CA Telon PDS UTILITY –SEARCH ELEMENT SUMMARY                    PAGE  01

   3                            4                                      2   XREFEDIT
 MATCH               SEARCH ELEMENT DEFINITION
 COUNT
 ————    :---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----:
 0030      FIND XFER-TASK
 0002      FIND EMPL-KEY    WORD
 0008      FIND TASK        PREFIX
 0003      FIND TIME-INDEX  WORD
 0051      FIND ID          SUFFIX
 0003      FIND 'INDEXED-BY'
```

```
 MATCH               SEARCH ELEMENT DEFINITION
 COUNT
 ————    :---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----:
 0004              03  TRGEMPL-AREA           PIC X(600).
 0002                  05  EMPL-KEY.
 0048                      10  EMPL-ID        PIC X(6).
 0017                  05  EMPL-NAME          PIC X(25).
 0008                  05  EMPL-DOB           PIC 9(6).
```

Detail information

The following numbers correspond to the numbers on the preceding report.

Number	Indicates
1	Date of the execution of the report.
2	Function that you requested by using the FUNC parameter.
3	Number of times a specified Element Definition was found on the source libraries you supplied.
4	Control cards used.

Creating Customized Reports

The CA Telon Automated Documentation Facility creates three extract files. Two of these files are used by Autodoc itself in the creation of the standard reports. The third file provides additional information regarding CA Telon TDF entities. All three files can be used to create additional special, customized reports at your installation.

The earlier subject "Extract Files" describes how you can specify what should be contained in these files, and how the files are created.

This current subject gives you a detailed description of the layout of the Automated Documentation extract files so you can design additional reports as needed.

Custom Code Extract File: TNXCUST

Contains information regarding any TDF Entity that references a custom code member. Only Entities matching the selection criteria on the Extract Control Card will appear in this file, as shown next.

FIELD NAME	POS	LTH	DESCRIPTION
EXT-CUSTCODE	1	8	Custom code member name
EXT-ENTITY	9	10	Name of Entity referencing Custom Code member: Header, ID, and Type
EXT-HEADER	19	5	Header for referencing Entity
EXT-ENTITY-TYPE	24	2	Type of Referencing Entity
EXT-STATEMENT	26	7	CA Telon Statement where Custom Code member referenced within entity
EXT-PARAMETER	33	7	CA Telon parameter within Statement where Custom Code member referenced
EXT-SOURCE	40	3	Location where Custom Code member found: TDF, PDS, or UNK
EXT-WHERE-FOUND	43	2	Type of Entity actually containing TDF Custom Code member (blank if EXT-SOURCE=PDS)
EXT-SIZE-IN-LINES	45	4	Number of lines in TDF Custom Code member (blank if EXT-SOURCE=PDS)
EXT-DATE-LAST-UPDATED	49	6	Date that the TDF Custom Code member was last updated (blank if EXT-SOURCE=PDS)
EXT-LAST-UPDATE-USER	55	10	Last user to update the Custom Code member (blank if EXT-SOURCE=PDS)

Data Access Extract File: TNXDBDSEG

Contains information regarding Data Access references within Entities that are on the CA Telon TDF. Only Entities matching the selection criteria on the extract control card will appear in this file, as shown next.

FIELD NAME	POS	LNTH	DESCRIPTION
EXT-ENTITY	1	10	Complete name of Entity: Header, ID, and Type
EXT-HEADER	11	5	Header for Entity
EXT-ENTITY-TYPE	16	2	Entity Type
EXT-ACCESS-TYPE	18	1	Access Type
EXT-DBDNAME	19	8	DBD Name, or LTERM if a TPPCB, or Table name for DB2
EXT-PCBNAME	27	12	PCB Name
EXT-PROCSEQ	39	8	Processing sequence value
(TPPCB INFORMATION)			
EXT-EXPRESS	47	1	EXPRESS parameter value
EXT-ABCALL	48	1	ABCALL parameter value
EXT-PRINT	49	1	PRINT parameter value
(GENERAL DATA SET INFORMATION)			
EXT-CTLCHAR	50	1	ANSI control character used
EXT-OPEN-TYPE	51	1	Type of open requested in Q-1000-PROGRAM-INIT
EXT-BLKSIZE	52	5	Blocksize of file
EXT-LRECL-MIN	57	5	Minimum logical record length
EXT-LRECL-MAX	62	5	Maximum logical record length
(VSAM DATA SET INFORMATION)			
EXT-VSAM-ORGANIZATION	67	1	VSAM organization (K=KSDS, R=Relative, E=ESDS)
EXT-VSAM-ACCESS	68	1	VSAM ACCESS (D=Dynamic, R=Random, S=Sequential)
EXT-INDEXOF	69	8	VSAM data set of which this is an index
(DL/I SEGMENT INFORMATION)			
EXT-SEQ-NUMBER	77	4	Sequence number of this segment
EXT-SEGNAME	81	8	Segment/Record/Row name
EXT-IMSKEY	89	8	IMS key to this segment
EXT-USAGE	97	1	Segment usage
Filler	98	1	
EXT-COPY	99	8	Copy Code associated with this segment

FIELD NAME	POS	LNTH	DESCRIPTION
	(AUTOEXEC DATA ACCESS COUNTS)		
EXT-AUTO-CREATE-COUNT	107	3	Number of Autoexec Creates
EXT-AUTO-UPDATE-COUNT	110	3	Number of Autoexec Updates
EXT-AUTO-DELETE-COUNT	113	3	Number of Autoexec Deletes
EXT-OUTREAD-COUNT	116	3	Number of Autoexec Outreads
EXT-OIREAD-COUNT	119	3	Number of Autoexec OIreads
EXT-INREAD-COUNT	122	3	Number of Autoexec Inreads
EXT-BROWSE-COUNT	125	3	Number of Autoexec Browses
EXT-TRANSACT-COUNT	128	3	Number of Autoexec Transacts
	(USEREXEC DATA ACCESS INFORMATION)		
EXT-CREATE-COUNT	131	3	Number of Userexec Creates
EXT-UPDATE-COUNT	134	3	Number of Userexec Updates
EXT-DELETE-COUNT	137	3	Number of Userexec Deletes
EXT-ERASE-COUNT	140	3	Number of Userexec Erases
EXT-READ-COUNT	143	3	Number of Userexec Reads
EXT-READNEXT-COUNT	146	3	Number of Userexec Readnexts
	(SEGEDIT INFORMATION)		
EXT-SEGEDIT-COUNT	149	3	Number of Segedit references
	(OTHER)		
EXT-UIO-INQUIRY-ONLY	152	1	Y/N indicator if this Segment is read only
EXT-WRITE-COUNT	153	3	Number of writes against this segment
EXT-POSITION-COUNT	156	3	Number of positions against this segment
EXT-REPLACE-COUNT	159	3	Number of replaces against this segment
EXT-WORKSPA-COUNT	162	3	Number of WORKSPAs against this segment
EXT-HOLD-COUNT	165	3	Number of Holds against this segment
	(DB2 SPECIFIC INFORMATION)		
EXT-SQL-TABLE-QUAL	168	8	SQL qualifier name
EXT-SQL-TABLE-NAME	176	18	SQL Table or Join name
EXT-SQL-TYPE	194	1	Indicator of Table or Join (T=Table, J=Join)
FILLER	195	6	Reserved for future use

Entity Extract File: TNXENT

Contains information regarding Entities that are on the CA Telon TDF. Only Entities matching the selection criteria on the extract control card will appear in this file, as shown next.

FIELD NAME	POS	LNTH	DESCRIPTION
EXT-ENTITY	1	10	Complete name of Entity: Header, ID, and Type
EXT-HEADER	11	5	Header for Entity
EXT-ID	16	5	Entity ID
EXT-ENTITY-TYPE	21	2	Entity Type
EXT-STATUS	23	1	Entity Type as one character
EXT-LAST-USER	24	8	ID of last user to update entity
EXT-CREATE-DATE	32	6	Date entity was created
EXT-UPDATE-DATE	38	6	Date entity was last updated
EXT-UPDATE-TIME	44	4	Time entity was last updated
FILLER	48	6	Reserved for future use
EXT-DESCRIPTION	54	40	Description of Entity
FILLER	94	14	Reserved for future use

Chapter 7: Century Date Conversion Utility

The Century Date Conversion Utility provides the means by which you can update the date fields in panels and reports to accommodate Year 2000 dates.

The utility reads through a transport file, looking for FIELD objects with FLDTYPE or OFLDTYPE properties associated with them. If the FLDTYPE or OFLDTYPE is one of the Telon-supplied date field edits, or a client's USREDIT date field edit (identified in a conversion table), the utility calculates whether there is enough space to expand the field by two characters to accommodate the additional century digits.

If the CONVERT runtype is selected, and if no expansion errors are encountered, the utility performs the actual field conversions by increasing the field length by two bytes (for example, from 6 to 8 or 8 to 10), changing the field's starting position (if necessary), and changing the field's FLDTYPE and/or OFLDTYPE from its current value to the century-date equivalent (for example, from DATE to CDATE). Regardless of runtype, the utility produces a report that lists each date field, its new (for a successful CONVERT) or proposed position, length and FLDTYPE, and all associated DBNAMEs.

Note: Since CA Telon does not have access to or control over the DBNAME fields to/from which Telon screen fields are mapped, the DBNAME fields identified in the report must be modified manually to match any screen fields modified by the utility.

The Telon-supplied field edits which the utility converts or reports on are:

- DATE (converts to CDATE)
- JULIAN (converts to CJULIAN)
- INTLDT (converts to INTCDT)
- INTLJUL (converts to INTCJUL)

Besides the supplied field edits, the conversion utility converts any user date fields edits identified in a conversion file (called USREDITS on the mainframe). The format of this file is as follows:

Columns	Contents
1-8	Name of date edit to be converted from
9-10	(Not used)

Columns	Contents
11-18	Name of date edit to be converted to

The utility validates all table entries to ensure that both a convert-from and convert-to user edit name is listed for each entry.

The Century Date Conversion Utility attempts to expand date fields in three ways (in this order):

- By adding two bytes to the end of the existing field (such as, changing the length but not the position)
- By adding two bytes to the beginning of the existing field (such as, changing both the length and the position)
- By adding one byte to the beginning and one byte to the end of the existing field (such as, changing both the length and the position)

This section contains the following topics:

Executing the Century Date Utility

The JCL and PROC for executing the Century Date Utility are found in #pdsqual.INSTALL. These members are customized when you customize all your JCL during the install process. The PROC is in member TLNUCVDT and the JCL is in member JUCVDT. In the PWS, the Century Date Utility is the CNVTDATE template invoked under the options menu in the PWS Desktop.

In the JCL, identify the following:

- The source file that contains the transport source to be converted. This file may contain more than one Telon program. It may also contain other objects such as tables, files and panels. The utility ignores anything that is not relevant to the conversion.
- The source file where the converted transport source is to be placed, if CONVERT is selected and no errors are encountered.

  Note: File name must be supplied even if REPORT is selected. Alternatively, on the mainframe, the TRANOUT DD card in the CONVERT3 step of the TLNUCVDT JCL may be modified to //TRANOUT DD DUMMY; on PWS, you may reference a file called DUMMY in the CNVDATE invocation. The DUMMY file is supplied in the Telon root directory.

- The name of the USREDITS file that lists client user edits to be converted.

  Note: If you have no USEREDITS to identify for the conversion, you may "dummy out" this file. On the mainframe, change the three USREDITS DD cards (in steps CONVERT1, CONVERT2 and CONVERT3 of the TLNUCVDT JCL) to //USREDITS DD DUMMY. On PWS, you may reference a file called DUMMY in the CNVDATE invocation. The DUMMY file is supplied in the Telon root directory.

- The RUNTYPE: **REPORT** or CONVERT. If you specify REPORT, the utility performs all its tasks (such as, identifies all date fields, scopes out field expansion except the actual conversion. We recommend that you do a preliminary run in REPORT mode.

Century Date Conversion Utility Exception Conditions

During execution of the Century Date Conversion Utility, you may encounter one of the following exception conditions (displayed in SYSOUT on the mainframe and on the screen in PWS):

INPUT FILE CONTAINS NO DATE FIELDS TO CONVERT

This message is generated in the CONVERT1 step if the utility finds no date fields (with FLDTYPE or OFLDTYPE of the four Telon-supplied date edits or any edits specified in the USREDITS file) in the input transport file. This is not an error; it simply means there is nothing to convert. When this condition is encountered, the utility ends with a condition code of 22 in the CONVERT1 step.

MAXIMUM NUMBER OF FIELD EDITS EXCEEDED. ALL FIELD EDITS PAST 150 IGNORED.

This message is generated if the USREDITS conversion file contains more than 150 field edits. The maximum number of field edits allowed in CA Telon Generator is 150. This condition does not halt processing, but only the first 150 field edits listed will be reported on or converted by this utility.

FLDEDIT TABLE ENTRY BLANK; PROGRAM TERMINATED

This message is generated if either the convert-from or convert-to field edit is omitted from any of the entries in the USREDITS conversion file. This condition causes processing to halt.

INVALID PARM <parm>; "REPORT" ASSUMED

This message is generated if some value other than REPORT or CONVERT is specified as the RUNTYPE. If an invalid RUNTYPE is specified, the utility continues processing in REPORT mode.

The Century Date Conversion Status Report

The Century Date Conversion Status Report is produced whenever you run the Century Date Conversion Utility. The report provides different information, depending on whether it is run with Runtype Convert or Report.

- **Report** mode tells you whether you have enough space in the transport file field(s) to convert a date to a Year 2000 date, on a field-by-field basis.
- **Convert** mode performs the conversion, then provides information about each field converted. The conversion changes the transport file source whenever possible.

Note: The Century Date Conversion Utility is an "all or nothing" utility-either every applicable field must be able to be converted or no fields will be converted. That is, if a particular file contains five date fields but only four of the fields have enough space for the conversion, none of the fields are converted.

We recommend that you start by running in Report mode. This type of report tells you whether you can complete a conversion successfully, before you actually try to convert the date fields. If the status of the Report mode report is successful, the actual conversion should be successful.

Century Date Conversion Status Report Contents

The Century Date Conversion Status Report lists, on a panel-by-panel basis, the applicable fields and whether they can be converted automatically by the utility. The runtype parameter (report or convert) appears at the top of the report, including the status of the run (successful or unsuccessful).

Sample Report

A sample of the Century Date Conversion Status Report is shown next.

```
11                          CENTURY DATE CONVERSION STATUS REPORT
                            RUNTYPE: CONVERT  STATUS: UNSUCCESSFUL
                            1                 2

      3
    PANEL:  TRABCD
  FIELD     ** LN/COL/LTH  FLDTYPE                DBNAME:  9
  5         6    7         8
  -----     -- ----------  --------               ------

  DATE         01  63  10   CJULIAN               EIBDATE

  ADMDT     ** 05  12   8   DATE                  HOSP-ADMIT-DATE

  DISDT     ** 05  31   8   DATE                  HOSP-DISCHARGE-DATE


    PANEL:  TREFGH
  FIELD     ** LN/COL/LTH  FLDTYPE                DBNAME
  -----     -- ----------  --------               ------

  DATE         01   3  10   CDATE                XFER-TODAYS-DATE

  DOB          14  27  10   CDATE                EMPL-DOB

  DOE          16  27  10   CDATE                EMPL-DOE

4
    PANEL:  TRPV30        GROUP: TITLE
  FIELD     ** LN/COL/LTH  FLDTYPE                DBNAME
  -----     -- ----------  --------               ------

  DATE         01  88  10   CDATE                DATE()


    10
    DATES (IDENTIFIED ABOVE WITH **) CANNOT BE CONVERTED AS REQUESTED.  CONVERSION MUST BE DONE MANUALLY.
    WHEN CONVERTING THIS PROGRAM, BE SURE TO EXPAND ALL DBNAME FIELDS TO MATCH NEW DATE FIELD SIZES.
```

Detail Information

The following numbers correspond to the numbers on the preceding report.

Numbers	Description
1	**Runtype—**Specifies a Report or Convert. *Report* tells whether or not a field can be expanded and, if so, how. *Convert* performs the actual conversion, if it can be performed, and reports on what was done.
2	**Status—**Successful or Unsuccessful.
3	**Panel—**The name of the panel in which the date fields are found.
4	**Group—**The name of the panel group in which the date(s) are found. (Only displayed for reports)
5	**Field—**The name of the field on the panel.
6	****—**Indicates that this field cannot be converted.
7	**LN/COL/LTH—**Indicates the new field position and its length. For example, if converted, the Date field on panel TRABCD is found on Line 1, would begin in Column 63, and would be 10 characters in length. The existing line number, column, and length are displayed for those fields that cannot be converted.
8	**FLDTYPE—**The date value found in the FLDTYPE field. "C" in front of the value shown (for example, CJULIAN) indicates either a successful conversion (if Runtype is *Convert* and Status is *Successful*) or the potential conversion field type if the conversion is not done (if Runtype is *Report* or Status is *Unsuccessful*).
9	**DBNAME—**The DBNAME associated with the field where the information is mapped. **Note:** The CA Telon Century Date Utility cannot convert DBNAME fields. You are responsible for changing these fields to accommodate the converted date field sizes.
10	Messages pertaining to the report.

Appendix A: Sample Transport Data Administration Report

This appendix contains a sample of the CA Telon Transport Data Administration report which is produced on a Transport_In. The report lists details about the data administration objects (IMSDBD, DB2/SQL TABLE, DB2/SQL JOIN, CICSQUE, CICSJRNL, SEQUENTIAL or VSAM FILE) processed.

The Data Administration Report consists of three parts:

- The first part, shown on page 1 on the sample report, consists of the following information:
  - A summary list of objects
  - The results of the processing (under the STATUS heading)
  - The options specified on control cards
- The second part, shown on pages 2 through 11 on the sample report, lists one or more pages of detailed information for each object processed.

  This information includes differences between the data definition objects in the Transport file and the equivalent administration objects on the target TDF. Differences identified in this part of the report help you maintain data consistency between multiple TDFs. See Appendix D, *Managing Multiple TDFs,* for detailed information on maintaining a multiple TDF environment.
- The third part, shown on page 12 on the sample report, provides a summary of the number and type of objects processed.

This section contains the following topics:

Sample CA Telon Transport-Data Administration Report (see page 158)

Sample CA Telon Transport-Data Administration Report

Input control cards

The following input cards specify the options requested to generate the sample Data Administration Report:

```
TRANPORT IMPORTSELIGNORE DEVELOPMENT TRANSPORT TESTING                  00321300
TABLE    TELON.TRGEMPL                              MERGE  04          00321400
TABLE    TELON.TRGEMPL1                             MERGE  04          00321500
IMSDBD   TRGDBDV1                                   MERGE  04          00321600
IMSDBD   TRGDBDVX                                   MERGE  04          00321700
FILE     TRGEMPLV                                   MERGE  04          00321800
FILE     TRGEMPLA                                   MERGE  04          00321900
FILE     EMPMAST                                    MERGE  04          00322000
CICSQUE  EMPLQ                                      MERGE  04          00323000
CICSJRNL TNJOURN1                                   MERGE  04          00324000
END                                                                    00324100
```

Part 1

The first page of the sample report below provides the following information:

- The RUNTYPE specified on the TRANSPORT card, IGNORE, appears on the upper left side of the first page of the report in the TRANSPORT RUN TYPE field.
- The maximum severity, also specified on the TRANSPORT card, appears in the MAX ALLOWABLE SEVERITY field. The actual maximum severity encountered appears in the MAX SEVERITY THIS JOB field.
- The report lists each object type, identifier, status and the results of the job. As indicated under the STATUS heading, only two of the requested items were not transported.
- Under the COMMENTS heading, the report lists run options for each object. If no request for the object is specified in the control cards and it exists in the TRANSPORT file, a NOT TRANSPORTED message appears in the STATUS column without comments.

The control card specified for this sample is:

```
TRANPORT IMPORTSELIGNORE DEVELOPMENT TRANSPORT TESTING                    00321300
```

```
                                CA TELON TRANSPORT —— DATA ADMINISTRATION REPORT                    PAGE   1
                                                                                                    02/01/05
   CREATED BY JOB: TWEJO01G
   TRANSPORT RUN TYPE: IGNORE

   JOB RUN STATUS: MAX ALLOWABLE SEVERITY: 00
                   MAX SEVERITY THIS JOB:  12

   TRANSPORT STATUS:    TYPE            OBJECT NAME                        STATUS                 COMMENTS
                        ________  ____________________________________  ____________________  ____________________________
                        TABLE     TELON   TRGEMPL                      TRANSPORTED SUCCESSFULLY   OPTION = MERGE    MAX SEV = 04
                        TABLE     TELON   TRGEMPL1                     NOT TRANSPORTED            OPTION = MERGE    MAX SEV = 04
                        IMSDBD    TRGDBDV1                             TRANSPORTED SUCCESSFULLY   OPTION = MERGE    MAX SEV = 04
                        IMSDBD    TRGDBDVX                             NOT TRANSPORTED            OPTION = MERGE    MAX SEV = 04
                        FILE      TRGEMPLV                             TRANSPORTED SUCCESSFULLY   OPTION = MERGE    MAX SEV = 04
                        FILE      TRGEMPLA                             TRANSPORTED SUCCESSFULLY   OPTION = MERGE    MAX SEV = 04
                        FILE      EMPMAST                              TRANSPORTED SUCCESSFULLY   OPTION = MERGE    MAX SEV = 04
                        CICSQUE   EMPLQ                                TRANSPORTED SUCCESSFULLY   OPTION = MERGE    MAX SEV = 04
                        CICSJRNL  TNJOURN1                             TRANSPORTED SUCCESSFULLY   OPTION = MERGE    MAX SEV = 04

*****************************************************************************************************************************
```

Part 2

The second part of the sample report, which consists of pages 2 through 10, provides more details about the results of object processing, including:

- Maximum allowable severity value
- Maximum severity value encountered for the object
- Object name and action taken
- Any differences found during processing

If the system finds differences, the report lists the parameters that differ.

Note: The SOURCE heading refers to the TRANSPORT file, not the CA Telon source produced during export or COBOL or PL/I source.

Successful processing-SQL table

The next sample report page is an example of successful processing for an SQL table. The control card specified for this sample is:

```
TABLE     TELON.TRGEMPL                    MERGE  04
```

```
********************************************************************************************************************************
                                  CA TELON TRANSPORT —— DATA ADMINISTRATION REPORT                        PAGE   2
                                                                                                          02/01/05
   CREATED BY JOB: TWEJO01G
   TRANSPORT RUN TYPE: IGNORE

   OBJECT RUN STATUS:   MERGE PROCESSING PERFORMED
                        MAX ALLOWABLE SEVERITY: 04
                        MAX SEVERITY THIS OBJECT: 00
                        TELON   TRGEMPL                       MERGED


TELON    TRGEMPL            —— SQL TABLE ..... —— NO DIFFERENCES

********************************************************************************************************************************
```

Unsuccessful processing-SQL table

The sample report pages shown next provide an example of unsuccessful processing for an SQL table, as shown by the data under the DIFFERENCES FOLLOW message.

The control card specified for this sample is:

```
TABLE    TELON.TRGEMPL1                          MERGE  04
```

```
*****************************************************************************************************************
                                  CA TELON TRANSPORT —— DATA ADMINISTRATION REPORT                      PAGE   3
                                                                                                        02/01/05
   CREATED BY JOB: TWEJO01G
   TRANSPORT RUN TYPE: IGNORE

   OBJECT RUN STATUS:   MERGE PROCESSING PERFORMED
                        MAX ALLOWABLE SEVERITY: 04
                        MAX SEVERITY THIS OBJECT: 12
                        TELON   TRGEMPL1                    NOT MERGED


TELON    TRGEMPL1           —— SQL TABLE ..... —— DIFFERENCES FOLLOW

=============================================================================

  TRGEMPL1                  —— TELON ROW ..... —— **
    EMPL_ID                 —— COLUMN ..... —— **
    EMPL_NAME               —— COLUMN ..... —— **
    EMPL_DOB                —— COLUMN ..... —— **
    EMPL_SEX                —— COLUMN ..... —— **
    EMPL_PHONE              —— COLUMN ..... —— **
    EMPL_STREET             —— COLUMN ..... —— **
    EMPL_CITY               —— COLUMN ..... —— **
    EMPL_STATE              —— COLUMN ..... —— **
    EMPL_ZIP                —— COLUMN ..... —— **
    EMPL_DOE                —— COLUMN ..... NOT FOUND ON TDF ——                  SEVERITY —— 12
    EMPL_DEPARTMENT         —— COLUMN ..... —— **
    EMPL_HOURLY_RATE        —— COLUMN ..... —— **
    EMPL_HOURS              —— COLUMN ..... —— **
=============================================================================

  TRGEMPL                   —— TELON ROW ..... —— **
    EMPL_ID                 —— COLUMN ..... —— **
    EMPL_NAME               —— COLUMN ..... —— **
    EMPL_DOB                —— COLUMN ..... —— **
    EMPL_SEX                —— COLUMN ..... —— **
    EMPL_PHONE              —— COLUMN ..... —— **
    EMPL_STREET             —— COLUMN ..... —— DIFFERENCES FOLLOW


        PARAMETER         ACTION                          VALUES
        ——————————————    ————————————————————————————    ————————————————————————————————————————————————

        DCLCOL—NOT NULL                          SEVERITY 08 TDF    Y
                                                             SOURCE N

        DCLCOL—LENGTH                            SEVERITY 08 TDF    0025
                                                             SOURCE 0001


    EMPL_CITY               —— COLUMN ..... —— **
    EMPL_STATE              —— COLUMN ..... —— **
    EMPL_ZIP                —— COLUMN ..... —— **
    EMPL_DOE                —— COLUMN ..... —— **
    EMPL_DEPARTMENT         —— COLUMN ..... —— **
    EMPL_HOURLY_RATE        —— COLUMN ..... —— **
    EMPL_HOURS              —— COLUMN ..... —— **

=============================================================================

  IDCHECK                   —— TELON ROW ..... —— **
                                  CA Telon TRANSPORT —— DATA ADMINISTRATION REPORT                      PAGE   4
                                                                                                        02/01/05
   CREATED BY JOB: TWEJO01G
   TRANSPORT RUN TYPE: IGNORE

    EMPL_ID                 —— COLUMN ..... —— **

*****************************************************************************************************************
```

Successful processing-DL/I database

The sample report page shown next is an example of successful processing for a DL/I database. The control card specified for this sample is:

```
IMSDBD   TRGDBDV1                                    MERGE  04
```

```
IMSDBD   TRGDBDV1                                MERGE  04
*************************************************************************************************************
                                CA TELON TRANSPORT -- DATA ADMINISTRATION REPORT                    PAGE   5
                                                                                                    02/01/05
   CREATED BY JOB: TWEJ001G
   TRANSPORT RUN TYPE: IGNORE

   OBJECT RUN STATUS:   MERGE PROCESSING PERFORMED
                        MAX ALLOWABLE SEVERITY: 04
                        MAX SEVERITY THIS OBJECT: 00
                        TRGDBDV1                            MERGED


TRGDBDV1                    -- DL/I DATA BASE .. -- NO DIFFERENCES

=============================================================================================

  TRGEMPL                   -- SEGMENT ......... -- **

    **DFLT**                -- DL/I SSA ........ -- **

    GREATER                 -- DL/I SSA ........ -- **

=============================================================================================

  TRGINDX                   -- LCHILD .......... -- **

=============================================================================================

  TRGTASK                   -- SEGMENT ......... -- **

    TRGPROJ                 -- SEARCH FIELD .... -- **

    **DFLT**                -- DL/I SSA ........ -- **

=============================================================================================

  TRGTIME                   -- SEGMENT ......... -- **

    **DFLT**                -- DL/I SSA ........ -- **

*************************************************************************************************************
```

Unsuccessful processing-DL/I database

The sample report page shown next is an example of unsuccessful processing for a DL/I database. The control card specified for this sample is:

```
IMSDBD   TRGDBDVX                                    MERGE  04
```

```
********************************************************************************************************************************
                                CA TELON TRANSPORT -- DATA ADMINISTRATION REPORT                        PAGE   6
                                                                                                        02/01/05
   CREATED BY JOB: TWEJO01G
   TRANSPORT RUN TYPE: IGNORE

   OBJECT RUN STATUS:   MERGE PROCESSING PERFORMED
                        MAX ALLOWABLE SEVERITY: 04
                        MAX SEVERITY THIS OBJECT: 12
                        TRGDBDVX                            NOT MERGED


TRGDBDVX                    -- DL/I DATA BASE .. -- DIFFERENCES FOLLOW

================================================================================

  TRGINDX                   -- SEGMENT ......... -- DIFFERENCES FOLLOW


       PARAMETER         ACTION                          VALUES
       ------------      ------------------------        ------------------------------------------------------------
       SEGLTH MAXBYTES                           SEVERITY 08 TDF    00000
                                                             SOURCE 00006

       KEYLTH                                    SEVERITY 12 TDF    00016
                                                             SOURCE 00006


    **DFLT**                -- DL/I SSA ........ -- **

================================================================================

  TRGEMPL                   -- LCHILD .......... -- **

********************************************************************************************************************************
```

Successful processing-VSAM data set

The sample report page shown next is an example of successful processing for a VSAM data set. The control card specified for this sample is:

```
FILE     TRGEMPLV                                    MERGE  04
```

```
********************************************************************************************************************************
                                  CA TELON TRANSPORT -- DATA ADMINISTRATION REPORT                        PAGE   7
                                                                                                         02/01/05
   CREATED BY JOB: TWEJO01G
   TRANSPORT RUN TYPE: IGNORE

   OBJECT RUN STATUS:   MERGE PROCESSING PERFORMED
                        MAX ALLOWABLE SEVERITY: 04
                        MAX SEVERITY THIS OBJECT: 00
                        TRGEMPLV                              MERGED


TRGEMPLV                    -- VSAM DATA SET ... -- **



********************************************************************************************************************************
```

Successful with differences-VSAM

The next sample report page is an example of successful processing, with differences, for a VSAM data set. Although the system encountered differences, they were not significant enough for processing to fail. The control card specified for this example is:

```
FILE     TRGEMPLA                                MERGE  04
```

```
********************************************************************************************************************
                                  CA TELON TRANSPORT -- DATA ADMINISTRATION REPORT                    PAGE   8
                                                                                                      02/01/05
   CREATED BY JOB: TWEJO01G
   TRANSPORT RUN TYPE: IGNORE

   OBJECT RUN STATUS:   MERGE PROCESSING PERFORMED
                        MAX ALLOWABLE SEVERITY: 04
                        MAX SEVERITY THIS OBJECT: 04
                        TRGEMPLA                              MERGED


TRGEMPLA                    -- VSAM DATA SET ... -- DIFFERENCES FOLLOW


        PARAMETER         ACTION                            VALUES
        ------------      ------------------------          ------------------------------------------------------
        DEFAULT KEYLTH    SOURCE VALUE USED         SEVERITY 04 TDF    000
                                                               SOURCE 006


********************************************************************************************************************
```

Successful processing-Sequential data set

The next sample report page illustrates successful processing of a sequential data set. The control card specified for this example is:

```
FILE     EMPMAST                                    MERGE  04
```

```
********************************************************************************************************************************
                                  CA TELON:TRANSPORT —— DATA ADMINISTRATION REPORT                        PAGE   9
                                                                                                          02/01/05
   CREATED BY JOB: TWEJO01G
   TRANSPORT RUN TYPE: IGNORE

   OBJECT RUN STATUS:   MERGE PROCESSING PERFORMED
                        MAX ALLOWABLE SEVERITY: 04
                        MAX SEVERITY THIS OBJECT: 00
                        EMPMAST                              MERGED


EMPMAST                     —— SEQ DATA SET .... —— **

********************************************************************************************************************************
```

Successful processing-CICS queue

The following sample report page illustrates successful processing of a CICS queue. The control card specified for this example is:

```
CICSQUE  EMPLQ                                      MERGE  04
```

```
********************************************************************************************************************************
                                  CA TELON TRANSPORT —— DATA ADMINISTRATION REPORT                        PAGE  10
                                                                                                          02/01/05
   CREATED BY JOB: TWEJO01G
   TRANSPORT RUN TYPE: IGNORE

   OBJECT RUN STATUS:   MERGE PROCESSING PERFORMED
                        MAX ALLOWABLE SEVERITY: 04
                        MAX SEVERITY THIS OBJECT: 00
                        EMPLQ                                MERGED


EMPLQ                       —— CICS QUEUE ..... —— **

********************************************************************************************************************************
```

Successful processing-CICS journal

The following sample report page illustrates successful processing of a CICS journal. The control card specified for this example is:

```
CICSJRNL TNJOURN1                                    MERGE  04
```

```
********************************************************************************************************************************
                                  CA TELON TRANSPORT —— DATA ADMINISTRATION REPORT                        PAGE  11
                                                                                                          02/01/05
   CREATED BY JOB: TWEJO01G
   TRANSPORT RUN TYPE: IGNORE

   OBJECT RUN STATUS:   MERGE PROCESSING PERFORMED
                        MAX ALLOWABLE SEVERITY: 04
                        MAX SEVERITY THIS OBJECT: 00
                        TNJOURN1                              MERGED


TNJOURN1                    —— CICS JOURNAL .... —— **

********************************************************************************************************************************
```

Part 3

The next sample report page lists summary information about the requested transport processing, including:

- Object types
- Processing status
- Statistics by object type
- Totals by object type

The control card specified for this example is:

```
END
*****************************************************************************************************************
                                CA TELON TRANSPORT -- DATA ADMINISTRATION REPORT                    PAGE  12
                                                                                                    02/01/05
   CREATED BY JOB: TWEJO01G
   TRANSPORT RUN TYPE: IGNORE

                                          SUMMARY OF TRANSPORT PROCESSING

                                          REPLACED      CHECKED       MERGED       REFRESHED
     OBJECT          IGNORED    BYPASSED  OK  FAIL     OK  FAIL     OK  FAIL     OK  FAIL     ADDED    TOTAL
   ---------------   -------    --------  --- ----     --- ----     --- ----     --- ----     -----    -----
   DL/I DATA BASES       0          0      0    0       0    0       1    1       0    0        0        2

   DB2 TABLES/JOINS      0          0      0    0       0    0       1    1       0    0        0        2

   SEQ. DATA SETS        0          0      0    0       0    0       1    0       0    0        0        1

   VSAM DATA SETS        0          0      0    0       0    0       2    0       0    0        0        2

   CICS QUEUES           0          0      0    0       0    0       1    0       0    0        0        1

   CICS JOURNALS         0          0      0    0       0    0       1    0       0    0        0        1


*****************************************************************************************************************
*****************************************************************************************************************

 --  "**" IN TDF COMPARISON COMMENTS INDICATES NO DIFFERENCES  --


*****************************************************************************************************************
****                            ************                              ************                        *****
*** E N D    O F    R E P O R T ************ E N D    O F    R E P O R T ************ E N D  O F  R E P O R T ****
****                            ************                              ************                        *****
*****************************************************************************************************************

================================================================================
```

Appendix B: Manage Multiple TDFs with Transport

CA Telon's automation of the data administration synchronization process is provided by facilitating the synchronization of the data definition information in the TDF's data administration.

This appendix describes how to use Transport in a multiple TDF environment. A multiple TDF environment requires consistent data access information in each TDF and synchronization of changes. The inheritance of data administration information by the data access defined in a program must be controlled when data administration information changes.

This section contains the following topics:

Data Administration Synchronization (see page 171)

Data Administration Synchronization

Transport allows you to control the distribution of data administration (Option 2) information across multiple, active TDFs. This information includes IMS DBDs, DB2/SQL tables and JOINs, and VSAM and SEQUENTIAL files.

Two things are necessary to accomplish data administration synchronization:

- You need to implement a *master* TDF to ensure that essential information is never lost or destroyed.
- You need to support the maintenance of data consistency across multiple TDFs.

  CA Telon's Transport facility provides many flexible features to establish this consistency.

  Controlling multiple TDFs requires using a master TDF and controlling the transport of information. For more information about the use of options to control the transport of information, see the chapter "Transport Control File."

Master TDF

If individual developers are allowed to develop their own DSCs and TLNROWs, the DBA should require them to follow a naming convention to ensure that multiple different DSCs/TLNROWs are not created with identical names. All subordinate TDFs should at least compare their information during transport up to the master TDF to ensure that the master TDF is the repository for all information and that all names are unique in the master TDF.

Transporting Between the Master and Subordinate TDFs

When transporting out data administration objects from a subordinate TDF to the master, use AUTOREF on export to ensure that all relevant data administration information is transported. Upon transporting in to the master TDF, use AUTOMERGE processing with a maximum severity level of 04 for IMS and 08 for DB2/SQL. This allows only new DSCs/TLNROWs to be added.

Allow only the addition of DSCs or TLNROWs on the subordinate TDFs to eliminate any problems in maintaining the integrity of your master TDF. If more significant changes are allowed to be made on the subordinate development TDFs, control over the synchronization of your development efforts might be lost.

It is quite possible that two developers could add information that overlaps or contradicts each other. If this happens, you would have to back out one of the changes and reapply it to synchronize the TDFs data definitions.

The AUTOMERGE process and setting maximum severity levels is described in more detail later in this appendix for various types of development you are likely to be using.

Summary

If you follow the guidelines stated previously, you will minimize problems and keep the data access definitions completely in synchronization. These guidelines are as follows:

- Maintain central control by the DBA using a master TDF
- Use a naming convention for DSCs and TLNROWs added on subordinate TDFs
- Use AUTOMERGE processing with a maximum severity level that allows only new DSCs and TLNROWs to be added to the master TDF

Data Administration Synchronization (Development Issues)

When planning development, your DBA will create a new TDF or use an existing one. This topic provides the steps and issues involved for either process.

New Development On a Clean TDF

Initially, the DBA must designate a *master* TDF on which to centralize all development for a project. The DBA must also designate empty subordinate TDF(s) where the programs are to be worked on.

Then, your DBA must create all initial data definitions on the master TDF. This may not always be required, as you can often use data definitions from previously existing projects. If these data definitions exist on a TDF other than the current master, you must transport them to the new master using the appropriate procedures described in this appendix.

Transporting Data Definitions

The next step is to bring all the necessary data definitions down to the subordinate TDFs where the actual development is to take place. This includes not only the obvious objects, but also such hidden items as IMS DBDs, SQL tables, or files referenced in PSBs or FILEGRPS, physical databases or secondary indexes for IMS DBDs, and SQL tables referenced in SQL JOINs.

To accomplish this, transport out the designated data administration definitions from the master TDF with a processing option of AUTOREF. Auto referencing ensures that the system will transport out of the master TDF all the data definitions you need, including the hidden ones.

Perform transport in to bring the above transported information into the subordinate TDF. As you are transporting to a clean TDF, there would be no existing data access information that can conflict with what you are bringing in. Therefore, use a Transport option of IGNORE that will not only add information to the TDF, but ignores the information of the Transport file where a like object of the same name exists on the TDF.

Program Development on the Subordinate TDF

You are now ready to proceed with program development on this subordinate TDF. If during the development process new DSCs or TLNROWs are added, you have to regularly move them up to the master TDF to retain a true master. As discussed previously, no changes greater than the addition of DSCs or TLNROWs must be permitted on the subordinate TDFs or you will no longer retain the data integrity you started with.

To move any new DSCs/TLNROWs from the development TDFs to the master, do one of the following:

- Transport-out the data definition objects explicitly and use AUTOREF.
- Transport-out the programs being worked on using AUTOREF and the system will also transport all the data definitions referenced in the programs.

IMS development

For IMS-only development, transport in with the AUTOMERGE processing option and a maximum severity level of 04 (AUTOM04). This allows new DSCs to be added to the master, but aborts Transport with any greater differences.

DB2/SQL development

For DB2-only development, run transport-in with the AUTOMERGE processing option and a maximum severity level of 08 (AUTOM08). This allows new TLNROWs to be added to the master, but aborts Transport with any greater differences.

IMS and DB2 development

For mixed IMS and DB2 development, you must run transport-in with AUTOM04 and an object override card of 'TABLE * MERGE08' because TLNROWs report a level 08 severity when they are added. By using the 08 on only the Table override, you will not allow any other objects to merge at this higher level.

Note: If information changes after development has begun on the subordinate TDF, see Maintenance on an Existing TDF for the relevant procedures.

New Development On an Existing TDF

When planning development on an existing TDF you must still have your DBA designate a *master* TDF that is to contain the reference versions of all data administration information. In this instance, the master TDF may be an already existing TDF that is the master for a previous project, or it may be a brand new TDF newly designated as the master for this effort.

Your DBA should then create all initial data definitions on the master TDF. Sometimes this may not be required, as you can often use data definitions from previously existing projects.

Transporting Data Definitions

However you get the data definitions, you want to add this information to the subordinate TDF on which the development is going to take place. Again this includes the hidden items discussed in the previous appendix and the specifically named objects.

Just as with a new TDF, you will transport-out the designated data administration definitions from the master TDF with a processing option of AUTOREF to ensure that all the data definitions that you need are transported out of the master TDF.

Then, you transport -in to bring the above transported information into the subordinate TDF. As you are transporting into a TDF that already contains data, there may be information conflicting between the Transport file and the subordinate TDF.

IMS development

For IMS-only development, use the AUTOMERGE processing option with a maximum severity level of 04 (AUTOM04). This allows new DSCs to be added to the master, but aborts Transport with any greater differences. (Adding new information may not be a problem).

IMS and DB2 development

For mixed IMS and DB2 development, you must run transport-in with AUTOM04 and an object override card of 'TABLE * MERGE08' because TLNROWs report a level 08 severity when they are added. By using the 08 on only the Table override, you will not allow any other objects to merge at this higher level.

DB2/SQL development

For DB2-only development, use the AUTOMERGE processing option and a maximum severity level of 08 (AUTOM08). This allows new TLNROWs to be added to the master, but aborts Transport with any greater differences.

If Transport encounters greater severities than those allowed, you must analyze the differences and make a decision on what to do. This could range from running Transport with a higher severity level to making manual updates to either the master or the subordinate TDF. A key consideration in any such decision must be retaining the integrity of the master TDF.

As an alternative you may want to run transport-in with the specific override option of CHECK for all the data definitions, before attempting the Automerge. The CHECK option only performs the comparison routines, and never brings in any data to the TDF. In this way, you can ensure that you know what is on the TDF before any attempts are made to bring in the objects to be worked on.

Program Development On the Subordinate TDF

You are now ready to proceed with program development on this subordinate TDF. If during the development process new DSCs or TLNROWs are added, you have to regularly move them up to the master TDF to retain a true master. As discussed previously, no changes greater than the addition of DSCs or TLNROWs must be permitted on the subordinate TDFs or you will no longer retain the data integrity you started with.

To move any new DSCs/TLNROWs from the development TDFs to the master, do one of the following:

- Transport-out the data definition objects explicitly and use AUTOREF.
- Transport-out the programs being worked on using AUTOREF and the system will also transport all the data definitions referenced in the programs.

IMS development

For IMS-only development, transport in with the AUTOMERGE processing option and a maximum severity level of 04 (AUTOM04). This allows new DSCs to be added to the master, but aborts Transport with any greater differences.

DB2/SQL development

For DB2-only development, run transport-in with the AUTOMERGE processing option and a maximum severity level of 08 (AUTOM08). This allows new TLNROWs to be added to the master, but aborts Transport with any greater differences.

IMS and DB2 development

For mixed IMS and DB2 development, you must run transport-in with AUTOM04 and an object override card of 'TABLE * MERGE08' because TLNROWs report a level 08 severity when they are added. By using the 08 on only the Table override, you will not allow any other objects to merge at this higher level.

Note: If information changes after development has begun on the subordinate TDF, see Maintenance on an Existing TDF for the relevant procedures.

Data Administration Synchronization (Maintenance Issues)

The idea of a master TDF is just as important to the maintenance of a program as it is to new development.

Maintenance on a Clean TDF

If you do not have a master TDF that contains both the program to be modified and the necessary data definitions, one must be created.

If you archive programs and data definitions somewhere other than a TDF, you need to bring the information to the designated master TDF.

For proper procedures to import definitions and IMS DBDs, see the JCL utilities information in the following documentation:

- *Installation Guide*
- *PWS Option Administration Guide*
- *Programming Concepts Guide*

Transporting Data Definitions

When you want to do maintenance on existing programs on a clean TDF, you must bring down not only the program that you will be working on but also all the data definitions that you need for this program. To accomplish this you can transport-out the program from a previously existing master TDF while using AUTOREF to output not only the program but all the data administration information associated with it.

If the master TDF does not contain the program that you will be maintaining, then you must separately bring in all the data administration information from the master TDF. AUTOREF of the applicable root objects will ensure that all objects are included that are needed. Then separately transport your program from its TDF so that it may also be transported down to the subordinate TDF.

To bring the above information from the Transport file into the subordinate TDF, you should perform a transport-in. Since you are transporting to a clean TDF there must be no existing data access information or programs that may conflict with what you are bringing in. Use a Transport option of IGNORE that only adds information to the TDF, but ignores the information on the Transport file if a like object of the same name exists on the TDF.

Program Development On the Subordinate TDF

You are now ready to proceed with program development on this subordinate TDF. If during the development process new DSCs or TLNROWs are added, you have to regularly move them up to the master TDF to retain a true master. As discussed previously, no changes greater than the addition of DSCs or TLNROWs must be permitted on the subordinate TDFs or you will no longer retain the data integrity you started with.

To move any new DSCs/TLNROWs from the development TDFs to the master, do one of the following:

- Transport-out the data definition objects explicitly and use AUTOREF.
- Transport-out the programs being worked on using AUTOREF and the system will also transport all the data definitions referenced in the programs.

IMS development

For IMS-only development, transport in with the AUTOMERGE processing option and a maximum severity level of 04 (AUTOM04). This allows new DSCs to be added to the master, but aborts Transport with any greater differences.

DB2/SQL development

For DB2-only development, run transport-in with the AUTOMERGE processing option and a maximum severity level of 08 (AUTOM08). This allows new TLNROWs to be added to the master, but aborts Transport with any greater differences.

IMS and DB2 development

For mixed IMS and DB2 development, you must run transport-in with AUTOM04 and an object override card of 'TABLE * MERGE08' because TLNROWs report a level 08 severity when they are added. By using the 08 on only the Table override, you will not allow any other objects to merge at this higher level.

Note: If information changes after development has begun on the subordinate TDF, see Maintenance on an Existing TDF for the relevant procedures.

Maintenance on an Existing TDF

This is the most complex of scenarios because you must move previously existing programs and data access definitions to a development TDF already containing programs or data definitions. However, the concept is still the same as with the other scenarios. The purpose is to maintain your data access information on a master TDF for consistency, while allowing developers the freedom to work on their own TDFs.

Your DBA must designate a specific TDF as the master and ensure that it contains the most current versions of the data definitions for the project.

Transporting Data Definitions

Run transport-out for both your program and the necessary data definitions whether they are in the same location or not. Both must be transported using AUTOREF to ensure that all the information is transported out of the master TDFs.

Perform transport-in to bring the above transported information into the subordinate TDF. As you are transporting into a TDF which already contains data, it is possible for information to conflict between the Transport file and the subordinate TDF.

IMS development

For IMS-only development, use the AUTOMERGE processing option with a maximum severity level of 04 (AUTOM04). This allows new DSCs to be added to the master, but aborts Transport with any greater differences. (Adding new information may not be a problem).

IMS and DB2 development

For mixed IMS and DB2 development, you must run transport-in with AUTOM04 and an object override card of 'TABLE * MERGE08' because TLNROWs report a level 08 severity when they are added. By using the 08 on only the Table override, you will not allow any other objects to merge at this higher level.

DB2/SQL development

For DB2-only development, use the AUTOMERGE processing option and a maximum severity level of 08 (AUTOM08). This allows new TLNROWs to be added to the master, but aborts Transport with any greater differences.

If Transport encounters greater severities than those allowed, you must analyze the differences and make a decision on what to do. This could range from running Transport with a higher severity level to making manual updates to either the master or the subordinate TDF. A key consideration in any such decision must be retaining the integrity of the master TDF.

As an alternative you can run transport-in with the specific override option of CHECK for all the data definitions before attempting the Automerge. The CHECK option only performs the comparison routines, and never brings in any data to the TDF. In this way, you can ensure that you know what is on the TDF before any attempts are made to bring in the objects to be worked on.

Program Development On the Subordinate TDF

You are now ready to proceed with program development on this subordinate TDF. If during the development process new DSCs or TLNROWs are added on this TDF, regularly move them up to the master TDF to retain a true master. No changes greater than the addition of DSCs/TLNROWs should be permitted on the subordinate TDFs or you no longer retain the data integrity you started with.

To move any new DSCs/TLNROWs from the development TDFs to the master, either transport-out the data definition objects explicitly and use AUTOREF, or transport-out the program(s) being worked on using AUTOREF and the system also transports all the data definitions referenced in the programs.

IMS development

For IMS-only development, run transport-in with the AUTOMERGE processing option and a maximum severity level of 04 (AUTOM04). This allows new DSCs to be added to the master, but aborts Transport with any greater differences.

DB2/SQL development

For DB2-only development, run transport-in with the AUTOMERGE processing option and a maximum severity level of 08 (AUTOM08). This allows new TLNROWs to be added to the master, but aborts Transport with any greater differences.

IMS and DB2 development

For mixed IMS and DB2 development, run transport-in with AUTOM04 and an object override card of "TABLE * MERGE08" because TLNROWs report a level 08 severity when they are added. By using the 08 on only the Table override, you do not allow any other objects to merge at this higher level.

Significant Change Considerations

If you want to make more significant changes than the addition of DSCs, do the changes only on the master TDF to avoid redundant or conflicting changes being made on the subordinate development TDFs.

After you have made the changes, you can decide the best course of action to propagate them to the development TDFs. This may be as simple as running transport-in with AUTOMERGE and a higher severity level, running transport-in with the REPLACE option to overlay the existing data definitions, to running with the CHECK option and making manual updates to the subordinate TDFS.

Whatever your decision, the overriding concern must always be to maintain a master TDF as the repository of the data definitions for the current project and for possible future use. For more information about Transport processing options, see the chapter, "Transport Control File." For more information about importing or data inheritance, see the *Programming Concepts Guide*.

Appendix C: Transport Installation

For CA Telon 2.1, 4, 5, and 5.1, the Transport system is distributed as part of the base product. Thus, if you have installed any of these, you have already installed the Transport system.

This appendix describes how to install the CA Telon Transport System on z/OS mainframe systems.

This section contains the following topics:

Customize Transport PROC/JOB

During z/OS installation, these procedures and job streams are customized according to site standards:

- TLNITRAN-Procedure
- TLNOTRAN-Procedure
- JUTRANI-Job stream
- JUTRANO-Job stream

If necessary, you can make changes to these entities. An example of each follows.

TLNITRAN

```
//*********************************************************************
//* THIS PROCEDURE TRANSPORTS-IN A TRANSPORT FILE INTO A CA TELON   *
//* DESIGN FACILITY.                          *
//*********************************************************************
//*
//TLNITRAN PROC TRNFILE='#PDSQUAL.TRPTFILE',   USERS TRANSPORT FILE
//      PDSQUAL='#PDSQUAL',        CA Telon PDS QUALIFIER
//      VSQUAL='#VSAMQUAL',        VSAM QUALIFIER
//      SORTLIB='#SORTLIB',        SORTLIB
//      SCRUNIT='#SCRUNIT',        SCRATCH UNIT, EG.SYSDA
//*
//*********************************************************************
//*  DELETE TEMPORARY DATA SETS TO MAKE PROC REUSABLE IN JOBSTREAM  *

//*********************************************************************
//*
//SCRATCH EXEC PGM=IEFBR14
//DELETE1 DD DSN=&.&INMSGS.,DISP=(MOD,DELETE),
 //       UNIT=&SCRUNIT.,SPACE=(TRK,(0))
 //DELETE2 DD DSN=&.&SRTMSGS.,DISP=(MOD,DELETE),
 //       UNIT=&SCRUNIT.,SPACE=(TRK,(0))
 //*
```

```
//*********************************************************************
//*   EXECUTE TRANSPORT                          *
//*********************************************************************
//*
//TRANPORT EXEC PGM=TELONTDF,REGION=1028K,DYNAMNBR=4,
 //      PARM='ADPADLS/DLS,TPIMDRVR,TNFDMDL3,N'
//STEPLIB DD DSN=&PDSQUAL..LOAD,DISP=SHR
//SYSOUT  DD SYSOUT=*
//TLTRACE DD SYSOUT=*
//TNSYSLIB DD DSN=&PDSQUAL..LOAD,DISP=SHR
//TNTDD  DD DSN=&VSQUAL..TNTDD,DISP=SHR
//TNTDDW  DD DSN=&VSQUAL..TNTDDW,DISP=SHR
//TNTDF  DD DSN=&VSQUAL..TNTDF,DISP=SHR
//TNTDFW  DD DSN=&VSQUAL..TNTDFW,DISP=SHR
//TNTDX  DD DSN=&VSQUAL..TNTDX,DISP=SHR
//TNTDXW  DD DSN=&VSQUAL..TNTDXW,DISP=SHR
//TNTCCL  DD DSN=&VSQUAL..TNTCCL,DISP=SHR
//TNCCIMSG DD DSN=&.&INMSGS.,DISP=(NEW,PASS),
 //      UNIT=&SCRUNIT.,SPACE=(TRK,(15,15)),
 //      DCB=(LRECL=210,BLKSIZE=6300,RECFM=FB)
 //TPTRAN  DD DSN=&TRNFILE.,DISP=SHARE,
 //      DCB=(LRECL=80,BLKSIZE=6160,RECFM=FB)
 //TPRPT  DD SYSOUT=*
//*
//SORT  EXEC PGM=SORT,PARM=',SIZE=MAX'
//SORTLIB DD DSN=&SORTLIB.,DISP=SHR
//SORTIN  DD DSN=&.&INMSGS.,DISP=(OLD,DELETE)
 //SORTOUT DD DSN=&.&SRTMSGS.,DISP=(NEW,PASS),
 //      UNIT=&SCRUNIT.,SPACE=(TRK,(30,03)),
 //      DCB=(LRECL=210,BLKSIZE=6300,RECFM=FB)
 //SORTWK01 DD UNIT=&SCRUNIT.,SPACE=(TRK,(19,19))
 //SORTWK02 DD UNIT=&SCRUNIT.,SPACE=(TRK,(19,19))
 //SORTWK03 DD UNIT=&SCRUNIT.,SPACE=(TRK,(19,19))
 //SYSOUT  DD SYSOUT=*
//SYSIN  DD DSN=&PDSQUAL..INSTALL(CIMPSRT),DISP=SHR
//*
//RPT  EXEC PGM=TPPCIRPT,COND=(4,LT)
 //STEPLIB DD DISP=SHR,DSN=&PDSQUAL..LOAD
//     DD DISP=SHR,DSN=&PDSQUAL..LOAD
//     DD DISP=SHR,DSN=&PDSQUAL..LOAD
//SYSPRINT DD SYSOUT=*
//TNCCIMSG DD DSN=&.&SRTMSGS.,DISP=(OLD,DELETE)
 //TNCCIRPT DD SYSOUT=*,DCB=RECFM=FBA
//*
//****************** END OF TLNITRAN *********************************
```

TLNOTRAN

```
//TLNOTRAN PROC TRNFILE='#PDSQUAL.TRPTFILE',   USERS TRANSPORT FILE
//      PDSQUAL='#PDSQUAL',         CA TELON PDS QUALIFIER
//      VSQUAL='#VSAMQUAL',         VSAM QUALIFIER
//*
//*********************************************************************
//*   EXECUTE TRANSPORT                        *
//*********************************************************************
//*
//TRANPORT EXEC PGM=TELONTDF,REGION=1028K,DYNAMNBR=4,
 //      PARM='ADPADLS/DLS,TPIMDRVR,TNFDMDL3,N'
//STEPLIB DD DSN=&PDSQUAL..LOAD,DISP=SHR
//SYSOUT  DD SYSOUT=*
//TLTRACE DD SYSOUT=*
//TNSYSLIB DD DSN=&PDSQUAL..LOAD,DISP=SHR
//TNTDD  DD DSN=&VSQUAL..TNTDD,DISP=SHR
//TNTDDW  DD DSN=&VSQUAL..TNTDDW,DISP=SHR
//TNTDF  DD DSN=&VSQUAL..TNTDF,DISP=SHR
//TNTDFW  DD DSN=&VSQUAL..TNTDFW,DISP=SHR
//TNTDX  DD DSN=&VSQUAL..TNTDX,DISP=SHR
//TNTDXW  DD DSN=&VSQUAL..TNTDXW,DISP=SHR
//TNTCCL  DD DSN=&VSQUAL..TNTCCL,DISP=SHR
//TNCCIMSG DD DUMMY,DCB=(LRECL=210,BLKSIZE=210,RECFM=FB)
 //TPTRAN  DD DSN=&TRNFILE.,DISP=(,CATLG,DELETE),
       UNIT=&SCRUNIT., =(TRK,(30,5),RLS),
 //      DCB=(LRECL=80,BLKSIZE=6160,RECFM=FB)
 //TPRPT  DD SYSOUT=*
//*
//******************* END OF TLNOTRAN *********************************
```

JUTRANI

```
#JOBCARD1
#JOBCARD2
#JOBPARM
#JOBROUTE
//*********************************************************************
//*****            TRANSPORT              *****
//*****   THIS JOB WILL TRANSPORT IN A MEMBER(S) FROM A    *****
//*****   TRANSPORT FILE INTO THE TDF.             *****
//*****   #PDSQUAL.TRPTFILE MUST EXIST BEFORE THIS JOB CAN   *****
//*****   BE RUN.                        *****
//*****     THIS DEPENDS ON THE CONTROL CARD SETTING.     *****
//*****       CONTROL CARD = IMPORTALLREPLACE/IGNORE    *****
//*****       CONTROL CARD = IMPORTSELREPLACE/IGNORE    *****
//*****     SEE CA TELON TRANSPORT SYSTEM REFERENCE MANUAL  *****
//*********************************************************************
//*
//TRANPORT EXEC TLNITRAN,TRNFILE='#PDSQUAL.TRPTFILE'
//TRANPORT.TPCTL DD *
TRANPORT IMPORTALLREPLACE
END
/*
//
```

CA Telon Parameters

Customize both the TLNITRAN and TLNOTRAN procedures by supplying CA Telon information and Transport information. Supply the following CA Telon parameters:

- PDSQUAL—Specifies the common high-level qualifier used for your CA Telon non-VSAM data sets.
- VSQUAL—Specifies the common high-level qualifier used for your CA Telon VSAM data sets.

Transport-specific Parameters

The TLNITRAN and TLNOTRAN procedures contain the following Transport-specific parameters:

- TRNFILE—Specifies the name of a default Transport file
- SCRUNIT—Indicates the scratch disk unit
- SORTLIB—Specifies the name of the library used by the SORT program
- CTLLIB—Specifies the name of the CA Telon library containing control cards used to execute TDF utilities

The JUTRANI and JUTRANO JCL contains two components that you must modify. You will be prompted to supply the name and disposition of the Transport file to be processed, and the Transport operations to be performed.

Each time Transport executes, you can modify the TRNFILE parameter as required.

Note: For transport-out processing, the TLNOTRAN procedure assumes the existence of the Transport file to which the system writes. If it does not exist then override the TPTRAN DD statement to supply UNIT/SPACE information.

The sample JUTRANI and JUTRANO JCL supply the transport control file as a SYSIN input file referenced as TPCTL.The sample JCL is set up to perform a transport-in of all objects on the Transport file. If the objects exist, they are replaced on the TDF.

For more information about the content of the transport control file, see the "Transport Control File."

Appendix D: Transport File Description Charts

The charts in this appendix describe the structure of the transport file for use by third-party vendors. Users who perform normal processing by using Transport to move information from one TDF to another need not know the layout of the transport file as described in this appendix.

Whenever this appendix refers to *information supplied*, it is referring to information that third-party vendors should supply. Inherited values are shown by the presence of an '@' in the proper byte of an object.

This appendix presents these topics:

- CA Telon object relation chart
- CA Telon record and reference charts

CA Telon Object Relation Chart

The CA Telon object relation chart lists the transport record header identifiers for all user-defined root and dependent objects supported in the Transport system. It does not list CA Telon internal objects because vendors should not be supplying the information contained in those objects.

If the root object has dependent objects, their transport record header's identifiers are listed in the sequence in which you must supply the dependent objects. For each level of nesting within a root object/dependent object, the chart indents dependent objects two spaces.

CA Telon Object Relation Chart

```
PANEL          Online Screen, Report/Batch Report Definition
 GROUP           Group Definition
   FIELD               Field Definition
         SRC             SRC Statement Definition
         XFEDIT          Cross Field Edit Definition
         SEGEDIT         Segment Edit Definition
PROGRAM        Online/Batch/IMS Driver/Report Program Def.
  DAREFOBJ      Data Group Reference Definition
    DARECORD          Segment/Row/Record Access Definition
       DAREQEST         User I/O Request Definition
         SQPARNTK    DB2 Extension Definition
         SQPARNT1    Fetch Details Definition
         SQPARNT2    DB2 User Datatypes Definition
         SQPARNT3    DB2 Get Diagnostics Definition
  PGMENV             Environment Parameters Definition
    MIDONLY           MFS MID Only Field Definition
  SORT               Batch Sort Definition
    SORTKEY           Batch Sort Key Definition
  MERGE              Batch Merge Definition
    MERGEGRP          Batch Merge Group Definition
      MERGEKEY          Batch Merge Key Definition
  SPPARAM            Stored Procedure Parameter Definition
  STPROCKY           Called Stored Procedure Program
  MATCHKEY           Match Key Definition
  LANGENV            Programming Language Parameters Definition
  CUSTCODE           Custom Code Member Definition
    CARDIMAG            Custom Code Member Data Lines
```

```
IMSDBD        IMS DBD Definition
  DATA SET       IMS Data Set Definition
  AREA          IMS Fast Path Area Definition
  SEGM          IMS Segment Definition
    DLIDSC        Segment Data Search Criteria Definition
          DSCQUAL       SSA Segment Search Qualification Def.
        SRCHFLD       IMS Field Definition
  LCHILD        IMS Logical Child Definition
    XDFLDIMS      XDFLD Definition

TABLE         DB2 Table Definition
  TLNROW        DB2 Row/Telon Row Definition
    DCLCOLDB2       Column Definition

JOIN          Joined DB2/TELON Table Definition
  JOINTBL       Joined Table Definition
  JOINCOL       Column Join Definition
  TLNROW        DB2 Row/Telon Row Definition
    DCLCOLDB2       Column Definition

FILE          Sequential/VSAM File Definition

CICSQUE       CICS Queue Definition

CICSJRNL      CICS Journal Definition

PSB           IMS PSB Definition
  PCB           IMS PCB Usage Definition
    SENSEG          Sensitive Segment Usage Definition

FILEGRP       File Group Definition
  PCB           IMS PCB Usage Definition
    SENSEG          Sensitive Segment Usage Definition
  SQL           DB2 Table/Joined Table Reference Definition
    SENROW          Sensitive Row Definition
  DATA SET       Data Set Reference Definition
  SENQUE        CICS Queue Reference Definition
  SENJRNL       CICS Journal Reference Definition

PSTORE        Presentation Store Definition
  PSHV          Presentation Store Value Definition

*             Comment Transport Record
```

Transport Record and Reference Charts

For each root and dependent object there is a record chart and a reference chart. These charts provide details of the objects on the Transport system.

Transport Record Chart Definitions

Each root and dependent object has a transport record chart to identify the transport records that are valid for a root or dependent object. If the root/dependent object has dependent objects, the chart lists their transport record identifiers. The chart uses these identifiers as pointers to other transport record chart items.

The transport record chart identifies the root or dependent object name in the heading of the chart. If the chart is for a dependent object, then the chart lists that object's parent object names preceding the dependent object name. The header/trailer transport records use the root/dependent object name. Also, the first transport record in the chart lists the root/dependent object name.

The transport records identified in the body of the chart supply the following information:

- **Transport Record Identifier**—An eight-character identifier assigned to the transport record.
- **Transport Record Type**—Identifies a transport record's contents. Values are:
  - **R**—Root object
  - **D**—Dependent object
  - **P**—Parameter
- **Data Length**—Identifies the length of the data value for the transport record.
- **Data Type**—Identifies the format of the data value for the transport record. Values are:
  - **A**—Alphanumeric, left justified
  - **N**—Numeric, right justified with leading zeros
- **Required Indicator**—Indicates whether you must supply a transport record or dependent object and whether a transport record has a generator default value associated with it. Values are:
  - Y—Yes; you must supply a transport record or dependent object
  - N—No; a transport record or dependent is not required
  - D—The transport record is associated with a generator default value

  You should not provide fields that are identified as defaulted unless they are required, or unless you need to override the default value. The associated transport reference chart entry identifies the defaulted value.

- **Maximum Number of Occurrences**—Identifies the number of times you can supply a transport parameter or dependent object.

  Note: The symbol **—**in the "Maximum Number of Occurrences" column means that there is no limit on this object.

- **FLDTYPE**—Identifies the CA Telon FLDTYPE edit performed on the transport parameter (for example, ALPHA, NONE, or NUMERIC).

Transport Reference Chart Definitions

Following each transport record hart is a transport reference chart. Each transport identifier corresponds to an object on the preceding record chart. Each entry contains descriptions for the listed transport parameter or dependent object. The descriptive text specifies special transport record definition rules or descriptions to clarify a record's contents. The reference chart can supply the following other references for a transport parameter:

- Statement/parameter reference—Identifies the CA Telon macro name and keyword (parameter) name associated with the transport record. See the Programming Concepts Guide. for descriptions of these macros and keywords.

  The chart also supplies a sub-list identifier for macro parameters contained in sub-lists.

- TDF screen/field reference—Identifies the CA Telon TDF screen and fields where a user enters the transport parameter on the TDF.

  Note: The chart can show multiple statements/parameter and/or TDF screen references.

PANEL Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
PANEL	R	6	A	Y	1	
KEY	P	6	A	Y	1	
DESC	P	40	A	Y	1	
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	A	N	1	NUMERIC
UPDTIME	P	4	N	N	1	NUMERIC
CRTDATE	P	6	N	N	1	NUMERIC
LASTUSED	P	6	N	N	1	NUMERIC

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
INPCHAR	P	1	A	Y+D	1	
OUTCHAR	P	1	A	Y+D	1	
OICHAR	P	1	A	Y+D	1	
SELCHAR	P	1	A	Y+D	1	
LITBRK	P	1	A	Y+D	1	
SIZELL	P	2	N	Y	1	NUMERIC
SIZECCC	P	3	N	Y	1	NUMERIC
UPRLWR	P	1	A	Y+D	1	
GROUP	D	8	A	N	240	

Transport identifier	Description
PANEL	PANEL DEFINITION. The PANEL Transport record identifies the type of Panel that is transported. Values are: ■ SCREEN—The Panel is an Online Report/Screen ■ BATCH—The Panel is a Batch Report
KEY	Values: ■ SP—SCREEN/ID ■ TDF—PANEL DEFINITION MENU/HEADER AND ID
DESC	SP-SCREEN/DESC
LASTUSER	TDF-LAST USER TO UPDATE PANEL
LASTUPD	TDF-PANEL LAST UPDATE DATE (YYMMDD)
UPDTIME	TDF-PANEL LAST UPDATE TIME (HHMM)
CRTDATE	TDF-PANEL CREATION DATE (YYMMDD)
LASTUSED	TDF-PANEL LAST USED DATE (YYMMDD)
INPCHAR	SP-PANEL/INPCHAR. The default value is '>'.
OUTCHAR	SP-PANEL/OUTCHAR. The default value is '<'.
OICHAR	SP-PANEL/OICHAR. The default value is '+'.
SELCHAR	SP-PANEL/SELCHAR. The default value is '|'.
LITBRK	SP-PANEL/LITBRK. The default value is '/'.
SIZELL	SP-PANEL/SIZE(LL)

Transport identifier	Description
SIZECCC	SP-PANEL/SIZE(CCC)
UPRLWR	SP-PANEL/UPRLWR. Values are: ■ U—All Literals are translated into uppercase ■ L—Translation does not occur
GROUP	GROUP DEFINITION; defines the groups of fields contained in the PANEL

PANEL/GROUP Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
GROUP	D	8	A	Y	1	
SIZELL	P	3	N	N	1	NUMERIC
USAGE	P	7	A	Y	1	
SKIPBEF	P	4	A	N	1	SKIP
SKIPAFT	P	4	A	N	1	SKIP
FMTCUST	P	8	A	N	1	NAME
PRINT	P	60	A	N	1	NAME
TDSKIP	P	2	N	N	1	NUMERIC
REPSEQ	P	42	N	N	1	NUMERIC
CTLVAR	P	60	A	N	1	
CTLLTH	P	3	N	N	1	NUMERIC
CTLPIC	P	30	A	N	1	
MINOR	P	8	A	N	1	
FORGRP	P	60	A	N	4	
SEGLOOP	P	5	A	N	1	
SEGUSAGE	P	8	A	N	1	
INCRE	P	28	N	N	1	NUMERIC
CINCRE	P	28	N	N	1	NUMERIC
REPEAT	P	28	N	N	1	NUMERIC
LINECNT	P	6	A	N	1	NAME
OCUST1	P	8	A	N	1	NAME

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
OCUST2	P	8	A	N	1	NAME
OCUST3	P	8	A	N	1	NAME
OSEGIDX	P	30	A	N	1	
SAVEKEY	P	30	A	N	7	
SAVETO	P	30	A	N	7	
ISEGIDX	P	30	A	N	1	
ICUST1	P	8	A	N	1	NAME
ICUST2	P	8	A	N	1	NAME
COLSGLP	P	1	A	N	1	YESNO
ICTLNM	P	6	A	N	1	
PAGE	P	1	A	N	1	YESNO
PAGESAV	P	2	N	N+D	1	NUMERIC
STBRKEY	P	60	A	N	1	OFNAME
SCHFLDC	P	60	A	N	1	NAME
SCHFLDI	P	8	A	N	1	NAME
SCHFLDL	P	3	N	N	1	NUMERIC
PAGEKEY	P	60	A	N	1	OFNAME
PKYUNIQ	P	1	A	N	1	YESNO
PKYLTH	P	3	N	N	1	NUMERIC
FIELD	D	8	A	N	99999	

Transport identifier	Description
GROUP	GROUP DEFINITION; SP—GROUP/LABEL. For BATCH PANELs, GROUPs are keyed by the name supplied on the GROUP Transport Record. You can define as many as 240 groups. When you exceed this limit, you receive the Key_Value of 'LIMIT REACHED' in the TPP04 Duplicate Key error message. For non—BATCH PANELs, each GROUP can only occur once. If you supply duplicate GROUPs, the USAGE (such as, TOPPAGE, DETAIL, and BOTPAGE) of the duplicated group is placed in the Key_Value. **Note:** Different Transport records within the GROUP Dependent Object are valid depending on whether the GROUP is for a BATCH or non—BATCH Panel. Transport Records with GROUP Macros (Statements/Parameters) associated with them are valid for BATCH Panels. Transport Records with SEGLOOP Macros associated with them are valid for non—BATCH DETAIL groups. No additional Transport Records other than the GROUP and USAGE Transport Records are valid for non—DETAIL non—BATCH Panels.
SIZELL	TDF—PANEL EDITORS/GROUP LINE —SIZE&semi. SIZELL Transport Record is required for BATCH Panels.
USAGE	SP—GROUP/USAGE&semi. values: ■ BATCH Panels: ■ 'TOPAGE' ■ 'TOPDTL' ■ 'DETAIL' ■ 'BOTPAGE' ■ 'CONTROL' ■ 'SUMMARY' ■ Non—BATCH Panels: ■ 'TOPPAGE' ■ 'DETAIL' ■ 'BOTPAGE' For non—BATCH Panels you define SEGLOOP Fields within the DETAIL Group. You define fields before the SEGLOOP in the TOPPAGE Group and you define Fields after the SEGLOOP in the BOTPAGE Group. If the PANEL does not contain a SEGLOOP, then only supply the TOPPAGE Group. You must define the Panel Level Consistency Edit Field in the TOPPAGE Group.

Transport identifier	Description
SKIPBEF	SP—GROUP/SKIPBEF
SKIPAFT	SP—GROUP/SKIPAFT
FMTCUST	SP—GROUP/FMTCUST
PRINT	SP—GROUP/PRINT
TDSKIP	SP—GROUP/TDSKIP
REPSEQ	SP—GROUP/REPSEQ
CTLVAR	SP—GROUP/CTLVAR
CTLLTH	SP—GROUP/CTLLTH
CTLPIC	SP—GROUP/CTLPIC
MINOR	SP—GROUP/MINOR
FORGRP	SP—GROUP/FORGRP. You can supply as many as four FORGRP Transport Records
SEGLOOP	SEGLOOP DEFINITION. TDF—PANEL DEFINITION MENU/SEGLOOP TYPE(FILE/TABLE)
SEGUSAGE	SP—SEGLOOP/USAGE
INCRE	SP—SEGLOOP/INCRE
CINCRE	SP—SEGLOOP/CINCRE
REPEAT	TDF—SEGLOOP/REPEAT COUNT VALUES
LINECNT	SP—SEGLOOP/LINECNT
OCUST1	SP—SEGLOOP/OCUST1
OCUST2	SP—SEGLOOP/OCUST2
OCUST3	SP—SEGLOOP/OCUST3
OSEGIDX	SP—SEGLOOP/OSEGIDX **Note:** In CA Telon 1.5, this parameter could be OF or INDEXED HVNAMES. This is no longer supported in CA Telon 2.0 and above.
SAVEKEY	SP—SEGLOOP/PAGEKEY(HVNAME1). You can supply as many as seven SAVEKEY Transport Records.
SAVETO	SP—SEGLOOP/PAGEKEY(HVNAME2). You can supply as many as seven SAVETO Transport Records. **Note:** SAVEKEY and SAVETO values are stored together internally. All occurrences of both fields cannot exceed a total of 254 bytes of data. Note also that in addition to the actual number of bytes supplied for each parameter, an additional byte is added for each parameter (except the last). If the values supplied are indexed names with intervening spaces, an

Transport identifier	Description
	additional two bytes are added.
ISEGIDX	SP—SEGLOOP/ISEGIDX
ICUST1	SP—SEGLOOP/ICUST1
ICUSTOM	■ SP—SEGLOOP/ICUST2 ■ SP—SEGLOOP/ICUSTOM In r2.0 and later releases, the ICUST2 parameter replaces the ICUSTOM parameter. ICUST2 contains the same values as the ICUSTOM parameter.
ICTLNM	SP—SEGLOOP/ICTLNM.
PAGE	SP—SEGLOOP/PAGE.
PAGESAV	SP—SEGLOOP/PAGESAV. The default value is 1.
STBRKEY	SP—SEGLOOP/STBRKEY.
SCHFLDC	SP—SEGLOOP/SCHFLDC.
SCHFLDI	SP—SEGLOOP/SCHFLDI.
SCHFLDL	SP—SEGLOOP/SCHFLDL.
PAGEKEY	SP—SEGLOOP/PAGEKEY.
PKYUNIQ	SP—SEGLOOP/PKYUNIQ. This parameter is valid only with the r2 CA Telon versions of Transport.
PKYLTH	SP—SEGLOOP/PKYLTH. This parameter is only valid with the r2 CA Telon version of Transport.
FIELD	FIELD DEFINITION. Identifies the fields associated with the GROUP.
COLSGLP	SP—SEGLOOP/COLSGLP. Allows you to specify if SEGLOOP should be displayed by rows (N) or by columns (Y). Only valid when CINCRE is specified.

PANEL/GROUP/FIELD Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
FIELD	D	8	A	Y	1	NAME
USAGE	P	2	A	Y	1	USAGE
POSLL	P	2	N	Y	1	NUMERIC

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
POSCCC	P	3	N	Y	1	NUMERIC
LTH	P	3	N	Y	1	NUMERIC
HELPMSG	P	25	A	N	1	LUPCASE
FLDTYPE	P	8	A	N	1	FLDTYPE
OFLDTYPE	P	8	A	N	1	FLDTYPE
DBNAME	P	60	A	N	3	OFNAME
OF	P	60	A	N	3	OFNAME
TEXT	P	60	A	N	5	
LITBRK	P	1	A	N	1	
REQ	P	1	A	N+D	1	REQ
PIC	P	28	A	N	1	LUPCASE
INIT	P	60	A	N	1	UPCASE
MAPOUT	P	60	A	N	1	NAME
FMTEXITN	P	3	N	N	1	
FMTEXITV	P	3	N	N	1	
FMTCNTL	P	1	A	N	1	
MAPBWS	P	1	A	N	1	YESNO
NEXTPGM	P	5	A	N	1	NEXTPGM
SCONSIS	P	8	A	N	1	NAME
INEDIT	P	1	A	N+D	1	YESNO
INDBIO	P	1	A	N+D	1	YESNO
SELKEYF	P	57	A	N	1	LUPCASE
SELKEYT	P	57	A	N	1	LUPCASE
ATTRPRO	P	1	A	N	1	YESNO
ATTRINT	P	6	A	N	1	ATTRINT
OUTATTR	P	1	A	N	1	YESNO
EACOLOR	P	2	A	N	1	EACOLOR
EAHIGH	P	2	A	N	1	EAHIGH
EAVALID	P	2	A	N	1	EAVALID

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
CNTGRP	P	9	A	N	1	NAME
CNTSCOPE	P	6	A	N	1	BCONVRT
CNTGROUP	P	8	A	N	1	NAME
TOTREF	P	8	A	N	1	LUPCASE
TOTSIZEL	P	2	N	N	1	NUMERIC
TOTSIZER	P	2	N	N	1	NUMERIC
CONVERT	P	62	A	N	4	
VALUES	P	62	A	N	4	
RANGE	P	62	A	N	4	
FORMAT	P	62	A	N	4	
IEXTEND	P	60	A	N	8	
OEXTEND	P	60	A	N	8	
CALC	P	62	A	N	4	
SRC	D	3	A	N	1000	
XFEDIT	D	6	A	N	-	
COLSGLP	P	1	A	N	1	YESNO
SEGEDIT	D	7	A	N	-	

Transport identifier	Description
FIELD	FIELD DEFINITION&semi. SP—FIELD/LABEL. The FIELD Transport record identifies the name of a field that is Transported, a field without a name, or the field as containing Panel Level consistency edits (SRC, XFEDIT, COLSGLP, or SEGEDITS). If you supply the data value '*PANEL', then you are defining Panel Level Consistency Edits (SRC, XFEDIT, COLSGLP, and SEGEDITS). If you are defining Panel Level Consistency Edits, then you are not required to input other Field Parameters. Any other value (or none) in the FIELD Transport Record defines a standard CA Telon field. FIELDS are keyed by the Field Name supplied on the FIELD Transport Record. Fields are also keyed by their Line and Column (POSLL/POSCCC) location. If duplicate Line/Columns are supplied, the Key_Value in the Duplicate Key Error Message contains the erroneous Line/Column position (the Line is a three digit value). Field Names (other than the Panel Level Consistency Edits Field) must start

Transport identifier	Description
	with an alphabetic character. Different rules apply for the remainder of the field name, depending on the target language that you use to generate the PANEL. If the target language is COBOL, then the remainder of the Field Name can contain alphabetic or numeric values. For PL/I, the Field Name can contain alphabetic, numeric, or valid National Characters (for example, &., %, #, $, @).
USAGE	SP—FIELD/USAGE.
POSLL	SP—FIELD/POSLL.
POSCCC	SP—FIELD/POSCCC.
LTH	SP—FIELD/LTH.
HELPMSG	SP—FIELD/HELPMSG.
FLDTYPE	SP—FIELD/FLDTYPE(EDIT-TYPE/IN-EDIT-TYPE).
OFLDTYPE	SP—FIELD/FLDTYPE(OUT-EDIT-NAME_.
DBNAME1	SP—FIELD/DBNAME. The first of three possible DBNAMEs for a field.
DBNAME2	SP—FIELD/DBNAME. The second of three possible DBNAMEs for a field.
DBNAME3	SP—FIELD/DBNAME. The third of three possible DBNAMEs for a field.
OF1	SP—FIELD/OF. The first of three possible OFs for a field.
OF2	SP—FIELD/OF. The second of three possible OFs for a field.
OF3	SP—FIELD/OF. The third of three possible OFs for a field.
TEXT	SP—FIELD/TEXT. You can supply as many as five TEXT Transport Records. The System uses the entire 60 bytes of the first four Transport Records. However, the System uses only the first 16 bytes of the fifth TEXT record.
LITBRK	SP—FIELD/TEXT. The LITBRK Transport Record identifies whether you ended a Literal field (Y) using a Literal Break Character or not (N).
REQ	SP—FIELD/REQ. The default value is 'N'.
PIC	SP—FIELD/PIC.
INIT	SP—FIELD/INIT.
MAPOUT	SP—FIELD/MAPOUT.
FMTEXITN	SP—FIELD/FMTEXIT(EXITNUM).
FMTEXITV	SP—FIELD/FMTEXIT(EXITVECT).
FMTCNTL	SP—FIELD/FMTCNTL. Values: Y—The FIELD is an MFS MIDONLY field and the FIELD Transport Record must contain a valid MFS field name of LTSEQ, LTNAME, TIME, DATE1, DATE2, DATE3, DATE4, LTPAGENO or LTMSG.

Transport identifier	Description
	N—The FIELD is a standard 3270-type field.
MAPBWS	TDF—UPDATE BATCH OUTPUT FIELD/BLANK.WHEN.SAME.
NEXTPGM	SP—FIELD/NEXTPGM.
SCONSIS	SP—FIELD/SCONSIS.
INEDIT	SP—FIELD/INEDIT. The default value is 'N'.
INDBIO	SP—FIELD/INDBIO. The default value is 'N'.
SELKEYF	SP—FIELD/SELKEY(HVNAME,HVNAME).
SELKEYT	SP—FIELD/SELKEY(SCREEN-KEY-FLDNAME, STORED-KEY-FLDNAME).
ATTRPRO	SP—FIELD/ATTRPRO.
ATTRINT	SP—FIELD/ATTRINT.
OUTATTR	SP—FIELD/OUTATTR.
EACOLOR	SP—FIELD/EACOLOR.
EAHIGH	SP—FIELD/EAHIGH.
EAVALID	SP—FIELD/EAVALID.
CNTGRP	SP—FIELD/CNTGRP.
CNTSCOPE	SP—FIELD/SCOPE.
CNTGROUP	SP—FIELD/SCOPE(GROUP NAME).
TOTREF	SP—FIELD/TOTREF.
TOTSIZEL	SP—FIELD/TOTSIZE(LD).
TOTSIZER	SP—FIELD/TOTSIZE(RD).
CONVERT	SP—FIELD/CONVERT. You can supply as many as four CONVERT Transport Records.
VALUES	SP—FIELD/VALUES. You can supply as many as four VALUES Transport Records.
RANGE	SP—FIELD/RANGE. You can supply as many as four RANGE Transport Records.
FORMAT	SP—FIELD/FORMAT. You can supply as many as four FORMAT Transport Records.
IEXTEND	SP—FIELD/IEXTEND. You can supply as many as eight IEXTEND Transport Records.
OEXTEND	SP—FIELD/OEXTEND. You can supply as many as eight OEXTEND Transport Records.
CALC	SP—FIELD/CALC. You can supply as many as four CALC Transport Records.

Transport identifier	Description
SRC	SRC DEFINITION. You can define the SRC, XFEDIT, and SEGEDIT Dependent objects in any sequence. **Note:** You can define SRC, XFEDITs, and SEGEDITs onlyfor Select fields or the dummy field used to define Panel Level Consistency Edits (*PANEL). You can supply a combined maximum of 999 SRC, SEGEDIT, and SEGEDIT Dependent Objects for a Field. When you exceed this limit, the Key_Value on the Duplicate Key Error Message is replaced with the value 'LIMIT REACHED'.
XFEDIT	XFEDIT DEFINITION.
COLSGLP	SP—SEGLOOP/COLSGLP
SEGEDIT	SEGEDIT DEFINITION.

PANEL/GROUP/FIELD/SRC Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SRC	D	GS	A	Y	1	

Transport identifier	Description
SRC	SRC/PROCEDURAL STATEMENT. You can supply any number of SRC statements before the SRC END Trailer Transport Record.

PANEL/GROUP/FIELD/XFEDIT Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
XFEDIT	D	8	A	Y	1	
DESC	P	64	A	N	1	
COND	P	60	A	N	5	
CURSOR	P	8	A	N	1	
ERRMSG	P	60	A	N	2	

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
HILIGHT	P	60	A	N	2	
ERRCHAR	P	60	A	N	1	
SEGLOOP	P	1	A	N	1	YESNO

Transport identifier	Description
XFEDIT	XFEDIT DEFINITION. SP—XFEDIT/LABEL
DESC	SP—XFEDIT DESCRIPTION.
COND	SP—XFEDIT/COND. You can supply as many as five COND Transport Records. The System uses the entire 60 bytes of the first four Transport Records. However, the System uses only the first 12 bytes of the fifth COND record.
CURSOR	SP—XFEDIT/CURSOR.
ERRMSG	SP—XFEDIT/ERRMSG. You can supply as many as two ERRMSG Transport Records.
HILIGHT	SP—XFEDIT/HILIGHT. You can supply as many as two HILIGHT Transport Records.
ERRCHAR	SP—XFEDIT/ERRCHAR.
SEGLOOP	SP—XFEDIT SEGLOOP. Specify whether XFEDIT should be performed for every SEGLOOP line in one pass.

PANEL/GROUP/FIELD/SEGEDIT Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SEGEDIT	S	8	A	Y	1	
DESC	P	64	A	N	1	
PCBNAME	P	12	A	N	1	NAME
SEGNAME	P	8	A	N	1	
SEGKEY	P	60	A	N	1	

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
WHEN	P	60	A	N	5	
ERROR	P	8	A	N+D	1	
CURSOR	P	8	A	N	1	
ERRMSG	P	60	A	N	2	
HILIGHT	P	60	A	N	2	
ERRCHAR	P	60	A	N	1	
FUNC	P	8	A	N	1	NAME
OPCODE	P	5	A	N	1	
ALTSSA	P	1	A	N	1	YESNO
CMDCODE	P	4	A	N	1	
IOAREA	P	30	A	N	1	NBALPHA
SSALIST	P	60	A	N	1	
SEGLTH	P	60	A	N	1	
GENKEYL	P	30	A	N	1	
INHERIT	P	1	A	N	1	
SEGLOOP	P	1	A	N	1	YESNO

Transport identifier	Description
SEGEDIT	SEGEDIT—FIELD EDIT DEFINITION&semi. SP—SEGEDIT/LABEL.
DESC	SP—SEGEDIT DESCRIPTION.
PCBNAME	SP—SEGEDIT/PCBNAME.
SEGNAME	SP—SEGEDIT/SEGNAME.
SEGKEY	■ SP—SEGEDIT/SEGKEY. ■ SP—SEGEDIT/KEY. The SEGEDIT/KEY parameter is a new r2 parameter that replaces the SEGKEY parameter. It contains the same values as the SEGKEY parameter.
WHEN	SP—SEGEDIT/WHEN. You can supply as many as four WHEN Transport Records.
ERROR	SP—SEGEDIT/ERROR. Values:

Transport identifier	Description
	■ 'FOUND' ■ 'NOTFOUND' (default)
CURSOR	SP—SEGEDIT/CURSOR.
ERRMSG	SP—SEGEDIT/ERRMSG. You can supply as many as two ERRMSG Transport Records.
HILIGHT	SP—SEGEDIT/HILIGHT. You can supply up to two HILIGHT Transport Records.
ERRCHAR	SP—SEGEDIT/ERRCHAR.
FUNC	SP—SEGEDIT/FUNC.
OPCODE	SP—SEGEDIT/OPCODE.
ALTSSA	■ SP—SEGEDIT/ALTSSA. ■ SP—SEGEDIT/QUALIFY. This parameter, new in r2,replaces the ALTSSA parameter. It contains the same values as the ALTSSA parameter.
CMDCODE	SP—SEGEDIT/CMDCODE.
IOAREA	SP—SEGEDIT/IOAREA.
SSALIST	SP—SEGEDIT/SSALIST.
SEGLTH	■ SP—SEGEDIT/SEGLTH. ■ SP—SEGEDIT/KEYLTH. This parameter, new for r2 and later, replaces the SEGLTH parameter. It contains the same values as the SEGLTH parameter.
GENKEYL	SP—SEGEDIT/GENKEYL.
INHERIT	Inheritance indicator for the SEGKEY parameter.
SEGLOOP	SP—SEGEDIT SEGLOOP. Specify whether SEGEDIT should be performed for every SEGLOOP line in one pass.

PROGRAM Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
PROGRAM	R	6	A	Y	1	
APPLID	P	2	A	N	1	

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
KEY	P	6	A	Y	1	
DESC	P	40	A	N	1	
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	N	N	1	NUMERIC
CRTDATE	P	6	N	N	1	NUMERIC
LASTUSED	P	6	N	N	1	NUMERIC
REMARKS	P	8	A	N	1	
NEXTPGM	P	5	A	N	1	
CURSOR	P	8	A	N	1	
SIZELL	P	3	N	N+D	1	NUMERIC
SIZECCC	P	3	N	N+D	1	NUMERIC
LANG	P	3	A	Y	1	
LANGLVL	P	4	A	N+D	1	
XFERWKA	P	60	A	N	5	
WKAREA	P	60	A	N	5	
OINIT1	P	8	A	N	1	
OINIT2	P	8	A	N	1	
CURSCUS	P	8	A	N	1	
OUTTERM	P	8	A	N	1	
PFKEYS	P	60	A	N	5	
ININIT1	P	8	A	N	1	
ININIT2	P	8	A	N	1	
FLDEDIT	P	8	A	N	1	
CONSIS	P	8	A	N	1	
INTERM	P	8	A	N	1	
SECTION	P	60	A	N	5	
PGMCUST	P	60	A	N	5	
HOLD	P	1	A	N+D	1	YESNO

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
HELP	P	1	A	N+D	1	YESNO
OUTIFIL	P	1	A	N	1	
CAPS	P	3	A	N+D	1	
EOFKEY	P	1	A	N	1	YESNO
REFRESH	P	1	A	N	1	YESNO
ALARM	P	1	A	N	1	YESNO
EATTR	P	1	A	N+D	1	YESNO
EAINC	P	7	A	N	1	EACOLOR
EAINH	P	5	A	N	1	EAHIGH
EAOUTC	P	7	A	N	1	EACOLOR
EAOUTH	P	5	A	N	1	EAHIGH
EALITC	P	7	A	N	1	EACOLOR
EALITH	P	5	A	N	1	EAHIGH
EAERRC	P	7	A	N	1	EACOLOR
EAERRH	P	5	A	N	1	EAHIGH
LINKWKA	P	8	A	N	1	
RPTDEST	P	8	A	N	1	
SELECT	P	8	A	N	1	
FILEDEF	P	8	A	N	1	
INIT	P	8	A	N	1	
GETTRAN	P	8	A	N	1	
PRCTRAN	P	8	A	N	1	
PARMS	P	40	N	N	1	NUMERIC
FRSTPGM	P	5	A	N	1	
XFER	P	8	A	N	1	
TERM	P	8	A	N	1	
UPDTA	P	1	A	N	1	
PRINTER	P	4	A	N	1	
INIT1	P	8	A	N	1	

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
INIT2	P	8	A	N	1	
INMAST	P	8	A	N	1	
INTRAN	P	8	A	N	1	
ENDTRAN	P	8	A	N	1	
MATCH	P	8	A	N	1	
TGREATR	P	8	A	N	1	
MGREATR	P	8	A	N	1	
STRUCTRE	P	1	A	N	1	
DAREFOBJ	D	7	A	N	99999	
SORT	D	8	A	N	21	
MERGE	D	1	A	N	20	
MATCHKEY	D	1	A	N	15	
PGMENV	D	10	A	N	2	
LANGENV	D	5	A	N	5	
CUSTCODE	D	8	A	N		

Transport identifier	Description
PROGRAM	PROGRAM DEFINITION. Supply the appropriate value to identify the type of program to be Transported. Values: ■ ONLINE—Online Screen/Report Programs ■ BATCH —Batch Report Programs ■ DRIVER—IMS Driver Programs ■ REPORT—Report Programs NONTERM—CICS Nonterminal Programs
APPLID	■ SP—SCREEN/APPLID ■ SP—BATCH/APPLID ■ SP—NONTERM/APPLID ■ SP—DRIVER/APPLID ■ SP—REPORT/APPLID

Transport identifier	Description
KEY	■ SP—SCREEN/HEADER and ID ■ SP—BATCH/HEADER and ID ■ SP—NONTERM/HEADER and ID ■ SP—DRIVER/HEADER and ID ■ SP—REPORT/HEADER and ID ■ TDF—ONLINE PROGRAM DEFINITION MENU/HEADER and ID ■ TDF—BATCH PROGRAM DEFINITION MENU/HEADER and ID
DESC	■ SP—SCREEN/DESC ■ SP—BATCH/DESC ■ SP—NONTERM/DESC ■ SP—DRIVER/DESC ■ SP—REPORT/DESC
LASTUSER	TDF—Last User to Update PROGRAM
LASTUPD	TDF—PROGRAM Last Update Date (YYMMDD)
UPDTIME	TDF—PROGRAM Last Update Time (HHMM)
CRTDATE	TDF—PROGRAM Creation Date (YYMMDD)
LASTUSED	TDF—PROGRAM Last Used Date (YYMMDD)
REMARKS	■ SP—SCREEN/REMARKS ■ SP—BATCH/REMARKS ■ SP—NONTERM/REMARKS ■ SP—DRIVER/REMARKS ■ SP—REPORT/REMARKS You can supply as many as five REMARKS Transport Records. The System uses the entire 60 bytes of the first four Transport Records. However, the System uses only the first 16 bytes of the fifth REMARKS record.
NEXTPGM	SP—SCREEN/NEXTPGM
CURSOR	SP—SCREEN/CURSOR
SIZELL	■ SP—SCREEN/SIZE(LL) ■ SP—BATCH/SIZE(LL)

Transport identifier	Description
	■ SP—NONTERM/SIZE(LL) ■ SP—REPORT/SIZE(LL) The programs have the following default values: ■ ONLINE = 24 ■ BATCH = 60 ■ REPORT = 55
SIZECCC	■ SP—SCREEN/SIZE(CCC) ■ SP—BATCH/SIZE(CCC) ■ SP—NONTERM/SIZE(CCC) ■ SP—REPORT/SIZE(CCC) The programs have the following default values: ■ ONLINE = 80 ■ BATCH = 133 ■ REPORT = 120
LANG	■ SP—SCREEN/LANG ■ SP—BATCH/LANG ■ SP—NONTERM/LANG ■ SP—DRIVER/LANG ■ SP—REPORT/LANG Values: ■ 'COB' ■ 'PLI'
LANGLVL	SP—TELON/LANGLVL. The default value is 2.0.
XFERWKA	■ SP—SCREEN/XFERWKA ■ SP—NONTERM/XFERWKA ■ SP—DRIVER/XFERWKA ■ SP—REPORT/XFERWKA You can supply as many as five XFERWKA Transport Records. The System uses the entire 60 bytes of the first four Transport Records. However, the System uses only the first 16 bytes of the fifth XFERWKA record.

Transport identifier	Description
WKAREA	■ SP—SCREEN/WKAREA ■ SP—BATCH/WKAREA ■ SP—NONTERM/WKAREA ■ SP—DRIVER/WKAREA ■ SP—REPORT/WKAREA You can supply as many as five WKAREA Transport Records. The System uses the entire 60 bytes of the first four Transport Records. However, the System uses only the first 16 bytes of the fifth WKAREA record.
OINIT1	■ SP—SCREEN/OINIT1 ■ SP—REPORT/OINIT1
OINIT2	■ SP—SCREEN/OINIT2 ■ SP—REPORT/OINIT2
CURSCUS	SP—SCREEN/CURSCUS
OUTTERM	■ SP—SCREEN/OUTTERM ■ SP—REPORT/OUTTERM
PFKEYS	SP—SCREEN/PFKEYS&semi. you can supply as many as five PFKEYS Transport Records. The System uses the entire 60 bytes of the first four Transport Records. However, the System uses only the first 16 bytes of the fifth PFKEYS record.
ININIT1	SP—SCREEN/ININIT1.
ININIT2	SP—SCREEN/ININIT2.
FLDEDIT	SP—SCREEN/FLDEDIT.
CONSIS	SP—SCREEN/CONSIS.
INTERM	SP—SCREEN/INTERM.
SECTION	■ SP—SCREEN/SECTION ■ SP—BATCH/SECTION ■ SP—NONTERM/SECTION ■ SP—DRIVER/SECTION ■ SP—REPORT/SECTION

Transport identifier	Description
	You can supply as many as five SECTION Transport Records. The System uses the entire 60 bytes of the first four Transport Records. However, the System uses only the first 16 bytes of the fifth SECTION record.
PGMCUST	■ SP—SCREEN/PGMCUST ■ SP—BATCH/PGMCUST ■ SP—NONTERM/PGMCUST ■ SP—DRIVER/PGMCUST ■ SP—REPORT/PGMCUST You can supply as many as five PGMCUST Transport Records. The System uses the entire 60 bytes of the first four Transport Records. However, the System uses only the first 16 bytes of the fifth PGMCUST record.
HOLD	■ SP—SCREEN/HOLD ■ SP—DRIVER/HOLD The default value is 'N'.
HELP	SP—SCREEN/HELP. The default value is N.
OUTIFIL	SP—SCREEN/OUTIFIL.
CAPS	SP—SCREEN/CAPS. The default value is ON.
EOFKEY	SP—SCREEN/EOFKEY.
REFRESH	SP—SCREEN/REFRESH.
ALARM	SP—SCREEN/ALARM.
EATTR	SP—SCREEN/EATTR. The default value is 'N'.
EAINC	SP—SCREEN/EAIN(COLOR).
EAINH	SP—SCREEN/EAIN(MODE).
EAOUTC	SP—SCREEN/EAOUT(COLOR).
EAOUTH	SP—SCREEN/EAOUT(MODE).
EALITC	SP—SCREEN/EALIT(COLOR).
EALITH	SP—SCREEN/EALIT(MODE).
EAERRC	SP—SCREEN/EAERR(COLOR).
EAERRH	SP—SCREEN/EAERR(MODE).
LINKWKA	SP—REPORT/LINKWKA.

Transport identifier	Description
RPTDEST	■ SP—BATCH/RPTDEST ■ SP—NONTERM/RPTDEST
SELECT	SP—BATCH/COBFCPY(MBRNAME1).
FILEDEF	SP—BATCH/COBFCPY(MBRNAME2).
INIT	■ SP—BATCH/INIT ■ SP—DRIVER/INIT
GETTRAN	■ SP—BATCH/GETTRAN ■ SP—NONTERM/GETTRAN
PRCTRAN	■ SP—BATCH/PRCTRAN ■ SP—NONTERM/PRCTRAN
PARMS	SP—BATCH/PARMS.
FRSTPGM	SP—DRIVER/FRSTPGM.
XFER	SP—DRIVER/XFER.
TERM	■ SP—BATCH/TERM ■ SP—DRIVER/TERM ■ SP—NONTERM/TERM
UPDTA	SP—SCREEN/UPDTA.
PRINTER	SP—NONTERM/PRNTDEST.
INIT1	■ SP—BATCH/INIT1 ■ SP—NONTERM/INIT1
INIT2	■ SP—BATCH/INIT2 ■ SP—NONTERM/INIT2
INMAST	SP—MATCH/INMAST.
INTRAN	SP—MATCH/INTRAN.
ENDTRAN	SP—MATCH/ENDTRAN.

Transport identifier	Description
MATCH	SP—MATCH/MATCH.
TGREATER	SP—MATCH/TGREATER.
MGREATER	SP—MATCH/MGREATER.
STRUCTRE	SP—BATCH/STRUCTRE.
DAREFOBJ	Data Groups. You can supply DAREFOBJ, PGMENV, LANGENV, and CUSTCODE Sub-Objects in any sequence.
SORT	SORT field definitions&semi. you can supply as many as 25 SORT definitions. If you exceed this limit, the Key_Value in the Duplicate Key Error Message contains 'LIMIT REACHED'.
MERGE	MERGE field definitions&semi. you can supply as many as 20 MERGE definitions. If you exceed this limit, the Key_Value in the Duplicate Key Error Message contains 'LIMIT REACHED.'
MATCHKEY	Match Key Definitions for PROGRAM.
PGMENV	Generated Environment Parameters&semi. you can supply PGMENV definitions for the IMS, TSO, CICS, and BATCH generated program environments.
LANGENV	Programming Language Parameters. You can supply as many as five different LANGENVs for each LANGENV Environment (for example, CICS, IMS, TSO, BATCH, and ALL).
CUSTCODE	PROGRAM Custom Code

PROGRAM/DAREFOBJ Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
DAREFOBJ	D	7	A	Y	1	
QUAL	P	8	A	N	1	NAME
TBLNAME	P	18	A	N	1	NAME
EXCLUDE	P	1	A	N	1	
DCLCOPY	P	8	A	N	1	NAME
DB2TYPE	P	5	A	N	1	NAME
SYNONYM	P	1	A	N	1	NAME
DBDNAME	P	8	A	N	1	NAME

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
PCBNAME	P	12	A	N	1	NAME
KEYLEN	P	3	N	N	1	NUMERIC
PROCSEQ	P	8	A	N	1	NAME
PROCOPT	P	4	A	N	1	ALPHA
NAME	P	8	A	N	1	
ACCESS	P	7	A	N	1	
DSORG	P	4	A	N	1	
REUSE	P	1	A	N	1	
BLKSIZE	P	5	N	N	1	NUMERIC
LRECL	P	5	N	N	1	NUMERIC
LRECLMAX	P	5	N	N	1	NUMERIC
OPEN	P	6	A	N	1	
INDEXOF	P	8	A	N	1	
EXPRESS	P	1	A	N	1	
ABCALL	P	1	A	N	1	
PRINT	P	1	A	N	1	
LTERM	P	8	A	N	1	NAME
MSGCALL	P	1	A	N	1	
JTYPEID	P	2	A	N	1	
JFILEID	P	2	N	N	1	
TYPE	P	2	A	N	1	
MAIN	P	1	A	N	1	
SCHEMA	P	8	A	N	1	
VERSION	P	4	N	N	1	NUMERIC
SSCHEMA	P	8	A	N	1	
DICTNAME	P	8	A	N	1	
NODE	P	8	A	N	1	
DBNAME	P	8	A	N	1	
DBNODE	P	8	A	N	1	

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
COMMIT	P	1	A	N	1	
FILENAM	P	60	A	N	1	
DEVASGN	P	40	A	N	1	
DARECORD	D	12	A	N	99999	

Transport identifier	Description
DAREFOBJ	■ SP—DATABAS/TYPE ■ SP—DATA SET/ACCESS(VSAM or SEQ) ■ TDF—UPDATE DATA GROUP/TYPE Supply the appropriate value to identify the type of Data Access Object being referenced. Valid values are: ■ 'DATABAS' or 'DB' ■ 'TPPCB' or 'TP' ■ 'GSAM' ■ 'DB2' or 'SQL' ■ 'QUEUE' ■ 'JOURNAL' ■ 'VSAM' ■ 'SEQ' ■ 'DATACOM' ■ 'SORT' ■ 'MATCH' ■ 'MERGE' You can supply a maximum of 999 DAREFOBJ's for a PROGRAM. If you exceed this limit, the Key_Value of the Duplicate Key Error Message is replaced with the value 'LIMIT REACHED'. **Note:** Different Transport records within the DAREFOBJ Dependent Object are valid depending on the type of DAREFOBJ you supply. Transport Records that have DATABAS Macros (Statement/Parameters) associated with them are valid for DAREFOBJ types of DATABAS and GSAM. Transport Records that have DATA SET Macros associated with them are valid for the DAREFOBJ types of VSAM, QUE, or SEQ. Transport

Transport identifier	Description
	Records that have TPPCB or CJOURNAL Macros associated with them are valid for the DAREFOBJ type of TPPCB or JRNL. Transport Records that have TABLE, DB2, DATA SET, and ROW Macros associated with them are valid for DAREFOBJ type of DB2, VSAM, or SEQ. Transport Records that have TPPCB Macros associated with them are valid for DAREFOBJ type of TPPCB. Transport Records that have TABLE, DB2, and ROW Macros associated with them are valid for DAREFOBJ type of DB2.
QUAL	■ SP—TABLE/QUAL ■ SP—DB2/TBLQUAL ■ SP—ROW/RBLQUAL
TBLNAME	■ SP—TABLE/NAME ■ SP—DB2/TBLNAME ■ SP—ROW/TBLNAME
EXCLUDE	Indicates whether or not you want to exclude this DAREFOBJ automatically from display. This automatic exclusion occurs when this PROGRAM's referenced Data Access Objects are edited in the Update Data Group Screen. Values are: ■ 'Y'—Exclude from view ■ 'N'—Display
DCLCOPY	SP—TABLE/DCLCOPY.
DB2TYPE	Identifies the type of DB2 Table for a DAREFOBJ of DB2. Values are: ■ 'TABLE'—Base DB2 Table ■ 'JOIN'—Joined DB2 Table
SYNONYM	■ SP—TABLE/SYNONYM ■ SP—DB2/SYNONYM
DBDNAME	■ SP—DATABAS/DBDNAME ■ SP—DLIDSC/DBDNAME
PCBNAME	■ SP—DATABAS/PCBNAME ■ SP—TPPCB/NAME
KEYLEN	SP—DATABAS/KEYLEN

Transport identifier	Description
PROCSEQ	SP—DATABAS/PROCSEQ
PROCOPT	SP—DATABAS/PROCOPT
NAME	SP—DATA SET/NAME
ACCESS	SP—DATA SET/ACCESS. Values are: ■ 'SEQ' ■ 'DYNAMIC' ■ 'RANDOM' ■ 'DIRECT' For a VSAM file, specify the access method you are using.
DSORG	SP—DATA SET/ACCESS. Values are: ■ 'KSDS' ■ 'RRDS' ■ 'ESDS' For a VSAM file, specify the file's internal organization.
REUSE	■ SP—DATA SET/REUSE ■ TDF—UPDATE DATA SET SEGMENT/REUSE
BLKSIZE	SP—DATA SET/BLKSIZE.
LRECL	■ SP—DATA SET/LRECL(REC-LENGTH OR MIN-REC-LENGTH) ■ SP—CQUEUE/LRECL ■ SP—CJOURNAL/LRECL
LRECLMAX	SP—DATA SET/LRECL(MAX-REC-LENGTH)
OPEN	SP—DATA SET/OPEN. Values are: ■ 'INPUT' ■ 'OUTPUT' ■ 'I-O' ■ 'EXTEND' ■ 'UPDATE'
INDEXOF	SP—DATA SET/INDEXOF.
EXPRESS	SP—TPPCB/EXPRESS.
ABCALL	SP—TPPCB/ABCALL.

Transport identifier	Description
PRINT	SP—TPPCB/PRINT.
LTERM	SP—TPPCB/LTERM.
MSGCALL	SP—TPPCB/MSGCALL.
JTYPEID	SP—CJOURNAL/JTYPID.
JFILEID	SP—CJOURNAL/JFILEID.
MAIN	SP—CQUEUE/MAIN.
TYPE	SP—CQUEUE/TYPE.
SCHEMA	SP—SCHEMA/name.
VERSION	SP—SCHEMA/VERSION.
SSCHEMA	SP—SCHEMA/SSCHEMA.
DICTNAME	SP—SCHEMA/DICTNAME.
NODE	SP—SCHEMA/NODE.
DBNAME	SP—DATABAS/DBDNAME.
DBNODE	SP—DATABAS/NODE.
COMMIT	SP—DATA SET/COMMIT.
FILENAM	SP—DATA SET/FILENAM.
DEVASGN	SP—DATA SET/DEVASGN.
DARECORD	Definitions. Use these records to define Sequential and VSAM records, DB2 Table Rows, or CA Telon Joined Rows.

PROGRAM/DAREFOBJ/DARECORD Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
DARECORD	D	12	A	Y	1	NAME
PARENT	P	8	A	N	1	NAME
LABEL	P	8	A	Y	1	NAME
USAGE	P	8	A	N	1	
ACALL	P	1	A	N	1	
KEY	P	60	A	N	4	OFNAME

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
INHERIT	P	7	A	N	1	
IGNORE	P	60	A	N	1	
COPY	P	20	A	N	1	NAME
COPYLV1	P	1	A	N	1	
COPYLBL	P	30	A	N	1	NAME
IMSKEY	P	8	A	N	1	NAME
KEYLEN	P	3	N	N	1	NUMERIC
KEYPIC	P	30	A	N	1	
OPCODE	P	60	A	N	1	NAME
CMDCODE	P	4	A	N	1	NAME
ALTSSA	P	6	A	N	1	NAME
IOAREA	P	60	A	N	1	
PROCOPT	P	5	A	N	1	
INDICES	P	60	A	N	1	
GENKEYL	P	30	A	N	1	
OPTLIST	P	60	A	N	1	
SEGLTH	P	60	A	N	1	
COBDIV	P	2	A	N	1	
COBVSKEY	P	60	A	N	1	
COBVSKYD	P	1	A	N	1	
WHERE	P	60	A	N	4	
TLNDSC	P	8	A	N	1	
EXCLUDE	P	1	A	N	1	
SYSID	P	4	A	N	1	
QUELBL	P	30	A	N	1	
LTHOPT	P	1	A	N	1	YESNO
LENGLBL	P	30	A	N	1	
JTYPEID	P	2	A	N	1	
RECNAME	P	16	A	N	1	NAME

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
RECVER	P	4	A	N	1	
AREANAME	P	16	A	N	1	NAME
SETNAME	P	16	A	N	1	NAME
READY	P	20	A	N	1	
LOCMODE	P	8	A	N	1	
LOGSSIND	P	1	A	N	1	
DAREQEST	D	8	A	N		

Transport identifier	Description
DARECORD	■ SP—SEGMENT/DBSEG ■ SP—RECORD/DBSEG You can supply as many as 999 DARECORDs for a DAREFOBJ. If you exceed this limit, the Key_Value in the Duplicate Key Error Message is replaced with the value 'LIMIT REACHED'. **Note:** Different Transport records within the DARECORD Dependent Object are valid depending on the type of DARECORD you supply. The type you supply is determined by the DAREFOBJ Dependent Object that the DARECORD is associated with. Transport Records that have SEGMENT Macros (Statement/Parameters) associated with them are valid for DAREFOBJ types of DATABAS and GSAM. Transport Records that have RECORD Macros associated with them are valid for DAREFOBJ types of QUE, JRNL, VSAM, or SEQ. Transport Records that have ROW Macros associated with them are valid for the DAREFOBJ type of DB2. Transport Records that have no Macros associated with them are valid for all DAREFOBJ types.
PARENT	SP—SEGMENT/PARENT.
LABEL	■ SP—SEGMENT/LABEL ■ SP—RECORD/LABEL ■ SP—ROW/LABEL
USAGE	■ SP—SEGMENT/USAGE ■ SP—RECORD/USAGE

Transport identifier	Description
	■ SP—ROW/USAGE
ACALL	This parameter exists only for r1.5 compatibility. If you supply a value, it is ignored. ■ 'Y' ■ 'N'
KEY	■ SP—SEGMENT/KEY ■ SP—RECORD/KEY ■ SP—ROW/KEY
INHERIT	Inheritance Indicators. Each byte of this record is used to indicate inheritance for the following parameters: ■ Byte 1—SEGKEY ■ Byte 2—COPY ■ Byte 3—COPYLV1 ■ Byte 4—COPYLBL ■ Byte 5—IMSKEY ■ Byte 6—OPCODE ■ Byte 7—CMDCODE **Note:** An inherited value is shown by the presence of an '@' in the proper byte of this object.
IGNORE	This parameter exists only for possible r1.5 compatibility. If you supply a value, it is ignored.
COPY	■ SP—SEGMENT/COPY ■ SP—RECORD/COPY
COPYLV1	■ SP—SEGMENT/COPYLV1 ■ SP—RECORD/COPYLV1
COPYLBL	■ SP—SEGMENT/COPYLBL ■ SP—RECORD/COPYLBL
IMSKEY	SP—SEGMENT/IMSKEY.
KEYLEN	■ SP—SEGMENT/KEYLEN

Transport identifier	Description
	■ SP—RECORD/KEYLEN
KEYPIC	SP—SEGMENT/KEYPIC.
OPCODE	■ SP—SEGMENT/OPCODE ■ SP—RECORD/OPCODE
CMDCODE	SP—SEGMENT/CMDCODE.
ALTSSA	This parameter exists only for possible r1.5 compatibility. If you supply a value, it is ignored.
IOAREA	This parameter exists only for possible r1.5 compatibility. If you supply a value, it is ignored.
PROCOPT	SP—SEGMENT/PROCOPT.
INDICES	SP—SEGMENT/INDICES.
GENKEYL	SP—RECORD/GENKEYL.
OPTLIST	SP—RECORD/OPTLIST.
SEGLTH	SP—RECORD/SEGLTH.
COBDIV	SP—RECORD/COBDIV.
COBVSKEY	SP—RECORD/COBVSKY(HVNAME).
COBVSKYD	SP—RECORD/COBVSKY(DUPLICATE).
WHERE	SP—ROW/WHERE. You can supply as many as four WHERE Transport Records.
TLNDSC	SP—SEGMENT/DSCREF. You can supply as many as four TLNDSC Transport Records.
EXCLUDE	Indicates whether you want to automatically exclude this DARECORD from display. This automatic exclusion occurs when this PROGRAM's DAREFOBJs are edited in the Update Data Group Screen. Values are: ■ Y—Exclude from view ■ N—Display
SYSID	SP—RECORD/SYSID.
QUELBL	SP—RECORD/QUELBL.
LTHOPT	SP—RECORD/LENGTH.
LENGLBL	SP—RECORD/LENGLBL.
JTYPEID	SP—RECORD/JTYPEID.
RECNAME	SP—RECORD/RECNAME.

Transport identifier	Description
RECVER	SP—RECORD/RECVERS.
AREANAME	SP—RECORD/AREA.
SETNAME	SP—RECORD/SET.
READY	SP—RECORD/READY.
LOCMODE	SP—RECORD/LOCMODE.
LOGSSIND	SP—RECORD/TYPE.
DAREQEST	User I/O Requests for the DARECORD.

PROGRAM/DAREFOBJ/DARECORD/DAREQEST Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
DAREQEST	D	8	A	Y	1	
LABEL	P	8	A	Y	1	NAME
FUNC	P	8	A	N	1	NAME
ALTSSA	P	1	A	N	1	
KEY	P	60	A	N	4	OFNAME
INHERIT	P	5	A	N	1	
IGNORE	P	60	A	N	1	
IOAREA	P	60	A	N	1	
OPCODE	P	60	A	N	1	NAME
CMDCODE	P	4	A	N	1	NAME
SSALIST	P	60	A	N	4	
PATH	P	8	A	N	1	NAME
PARENTG	P	8	A	N	1	NAME
LOCKED	P	8	A	N	1	NAME
CURRENT	P	8	A	N	1	NAME
CONCAT	P	1	A	N	1	
TLNDSC	P	8	A	N	15	
DSCSEG	P	8	A	N	15	

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
UPDATE	P	1	A	N	1	
DELETE	P	1	A	N	1	
COPYINIT	P	8	A	N	1	NAME
COPYCALL	P	8	A	N	1	NAME
COPYTERM	P	8	A	N	1	NAME
COPYKEY	P	8	A	N	1	NAME
EXCLUDE	P	1	A	N	1	
WHERE	P	60	A	N	4	
KEYCOLS	P	60	A	N	2	
DBMSNAME	P	16	A	N	1	NAME
SENCOLS	P	60	A	N	4	
ORDERBY	P	60	A	N	2	
GROUPBY	P	60	A	N	1	
HAVING	P	60	A	N	2	
GENKEYL	P	30	A	N	1	
OPTLIST	P	60	A	N	1	
RECLTH	P	60	A	N	1	
TPPARMS	P	60	A	N	4	
FDREC	P	30	A	N	1	
LTH	P	1	A	Y	1	YESNO
LENGLBL	P	30	A	N	1	
SET	P	1	A	N	1	YESNO
SETLBL	P	30	A	N	1	
NOSUSP	P	1	A	N	1	YESNO
ITEM	P	1	A	N	1	YESNO
ITEMLBL	P	30	A	N	1	
NUMITEM	P	1	A	N	1	YESNO
NITMLBL	P	30	A	N	1	
PFXLEN	P	2	N	N	1	NUMERIC

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
PFXLBL	P	30	A	N	1	
REQID	P	1	A	N	1	YESNO
REQIDLBL	P	30	A	N	1	
WAIT	P	1	A	N	1	YESNO
STARTIO	P	1	A	N	1	YESNO
JFUNC	P	4	A	N	1	
LOCKED	P	1	A	N	1	YESNO
QUALIFY	P	1	A	N	1	YESNO
WITHIN	P	1	A	N	1	
PROCVAR	P	30	A	N	1	
PROCCTL	P	30	A	N	1	
INDXKEY	P	30	A	N	1	
SYMBKEY	P	30	A	N	1	
ALTLOG	P	30	A	N	1	
TMPQUAL	P	8	A	N	1	
TMPNAME	P	18	A	N	1	
TMPCOMT	P	8	A	N	1	
OPTION	P	20	A	N	1	
DAPARNTK	D	1	A	N	99999	

Transport identifier	Description
DAREQEST	User I/O function. Valid values are: ■ BROWSE ■ CREATE ■ DELETE ■ ERASE ■ HOLD ■ INREAD ■ OIREAD

Transport identifier	Description
	■ OUTREAD ■ READ ■ READNEXT ■ REPLACE ■ TRANSACT ■ UPDATE ■ WORKSPA ■ ISCON DAREQESTs are keyed by the Label Transport Record's Data Value, which must be unique within the associated DARECORD. If duplicates are found, the Key_Value in the Duplicate Key Error Message contains the duplicate Label value. **Note:** For Macro/Keyword references below, the generic term "Data Access" is used for parameters that are valid for all User I/O Functions identified above. Different Transport Records within the DAREQEST Dependent Object are valid depending on the type of DBMS you are accessing. This is identified in the *Programming Concepts Guide*.
LABEL	■ SP—Data Access/LABEL ■ SP—DLIDSC/SEGLBL
FUNC	SP—DELETE/FUNC&semi. values: ■ SP—INREAD/FUNC ■ SP—OIREAD/FUNC ■ SP—OUTREAD/FUNC ■ SP—READ/FUNC ■ SP—READNEXT/FUNC ■ SP—TRANSACT/FUNC ■ SP—UPDATE/FUNC ■ SP—JOURNAL/FUNC
ALTSSA	This parameter exists only for possible r1.5 compatibility. If you supply a value, it is ignored.
KEY	SP—Data Access/KEY. You can supply as many as four KEY Transport Records.
INHERIT	Inheritance Indicators. Use each byte of this record to indicate inheritance

Transport identifier	Description
	for the parameters, as follows: ■ Byte 1—SEGKEY ■ Byte 2—IOAREA ■ Byte 3—OPCODE ■ Byte 4—CMDCODE ■ Byte 5—DLIDSC
IGNORE	SP—Data Access/IGNORE.
IOAREA	SP—Data Access/IOAREA.
OPCODE	■ SP—DELETE/OPCODE ■ SP—BROWSE/OPCODE ■ SP—ERASE/OPCODE ■ SP—INREAD/OPCODE ■ SP—OIREAD/OPCODE ■ SP—OUTREAD/OPCODE ■ SP—READ/OPCODE ■ SP—READNEXT/OPCODE ■ SP—TRANSACT/OPCODE ■ SP—UPDATE/OPCODE
CMDCODE	SP—Data Access/CMDCODE.
SSALIST	SP—Data Access/SSALIST. You can supply as many as four SSALIST Transport Records.
PATH	SP—Data Access/PATH.
PARENTG	SP—Data Access/PARENTG.
LOCKED	SP—Data Access/LOCKED.
CURRENT	SP—Data Access/CURRENT.
CONCAT	SP—Data Access/CONCATK.
TLNDSC	SP—Data Access/DSCREF(DLIDSC Name). You can supply as many as fifteen TLNDSC Transport Records. You must supply these Transport records in pairs with a matching DSCSEG Transport Record for each TLNDSC record supplied.
DSCSEG	■ SP—Data Access/DSCREF(Segment Name)

Transport identifier	Description
	■ SP—DLIDSC/SEGNAME You can supply as many as fifteen DSCSEG Transport Records. You must supply these Transport records in pairs with a matching TLNDSC Transport Record for each supplied.
UPDATE	SP—Data Access/UPDATE.
DELETE	SP—Data Access/DELETE.
COPYINIT	SP—Data Access/CPYINIT.
COPYCALL	SP—Data Access/CPYCALL.
COPYTERM	SP—Data Access/CPYTERM.
COPYKEY	SP—Data Access/CPYKEY.
EXCLUDE	Indicates whether you want to exclude this UIO automatically from display. This automatic exclusion occurs when this PROGRAM's DAREFOBJs are edited in the Update Data Group Screen. Values are: ■ Y—Exclude from view ■ N—Display
WHERE	■ SP—BROWSE/WHERE ■ SP—CREATE/GRPISRT ■ SP—DELETE/WHERE ■ SP—ERASE/WHERE ■ SP—INREAD/WHERE ■ SP—OIREAD/WHERE ■ SP—OUTREAD/WHERE ■ SP—READ/WHERE ■ SP—READNEXT/WHERE ■ SP—REPLACE/WHERE ■ SP—TRANSACT/WHERE ■ SP—UPDATE/WHERE You can supply as many as four WHERE Transport Records.
KEYCOLS	SP—Data Access/KEYCOLS. You can supply as many as four KEYCOLS Transport Records.
DBMSNAME	SP—Data Access/SEGNAME.
SENCOLS	SP—Data Access/SENCOLS. You can supply as many as four SENCOLS

Transport identifier	Description
	Transport Records.
ORDERBY	SP—Data Access/ORDERBY. You can supply as many as two ORDERBY Transport Records.
GROUPBY	SP—Data Access/GROUPBY.
HAVING	SP—Data Access/HAVING. You can supply as many as two HAVING Transport Records.
GENKEYL	■ SP—BROWSE/GENKEYL ■ SP—DELETE/GENKEYL ■ SP—ERASE/GENKEYL ■ SP—INREAD/GENKEYL ■ SP—OIREAD/GENKEYL ■ SP—OUTREAD/GENKEYL ■ SP—READ/GENKEYL ■ SP—READNEXT/GENKEYL ■ SP—UPDATE/GENKEYL ■ SP—TRANSACT/GENKEYL ■ SP—UPDATE/GENKEYL
OPTLIST	SP—Data Access/OPTLIST.
RECLTH	SP—Data Access/RECLTH.
TPPARMS	SP—CREATE/TPPARMS.
FDREC	■ SP—CREATE/FDREC ■ SP—DELETE/FDREC ■ SP—ERASE/FDREC ■ SP—READ/FDREC ■ SP—READNEXT/FDREC ■ SP—REPLACE/FDREC ■ SP—UPDATE/FDREC
LTH	SP—JOURNAL/LENGTH.
LENGLBL	SP—JOURNAL/LENGLBL.
SET	

Transport identifier	Description
	■ SP—CREATE/SET ■ SP—INREAD/SET ■ SP—OIREAD/SET ■ SP—OUTREAD/SET ■ SP—READNEXT/SET ■ SP—READ/SET
SETLBL	■ SP—CREATE/SETLBL ■ SP—INREAD/SETLBL ■ SP—OIREAD/SETLBL ■ SP—OUTREAD/SETLBL ■ SP—READNEXT/SETLBL ■ SP—READ/SETLBL
NOSUSP	■ SP—CREATE/NOSUSP ■ SP—INREAD/NOSUSP ■ SP—OIREAD/NOSUSP ■ SP—OUTREAD/NOSUSP ■ SP—READ/NOSUSP ■ SP—READNEXT/NOSUSP ■ SP—JOURNAL/NOSUSP
ITEM	SP—Data Access/ITEM.
ITEMLBL	SP—Data Access/ITEMLBL.
NUMITEM	SP—Data Access/NUMITEM.
NITMLBL	SP—Data Access/NITMLBL.
PFXLEN	SP—JOURNAL/PFXLEN.
PFXLBL	SP—JOURNAL/PFXLBL.
REQID	SP—JOURNAL/REQID.
REQIDLBL	SP—JOURNAL/REQIDLBL.
WAIT	SP—JOURNAL/WAIT.
STARTIO	SP—JOURNAL/STARTIO.
JFUNC	SP—JOURNAL/JFUNC.

Transport identifier	Description
LOCKED	SP—Data Access/LOCKED.
QUALIFY	■ SP—Data Access/QUALIFY
WITHIN	■ SP—Data Access/WITHIN
INDXKEY	SP—ISRETURN/INDXKEY.
SYMBKEY	SP—ISRETURN/SYMBKEY.
ALTLOG	SP—Data Access/ALTLOG.
TMPQUAL	SP—ROW/TMPQUAL
TMPNAME	SP—ROW/TMPNAME
TMPCOMT	SP—ROW/TMPCOMT
OPTION	■ SP—Data Access/OPTION ■ SP—ISACCEPT/OPTION ■ SP—ISRETURN/OPTION

PROGRAM/DAREFOBJ/DARECORD/DAREQUEST/DAPARNTK Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
DAPARNTK	D	1	A	Y	9999	
RECTYPE	P	1	A	Y	1	
CALCKEY	P	32	A	N	1	
CALCKYHV	P	60	A	N	1	
HVEXTND	P	1	A	N	1	
WHERE	P	60	A	N	1	

Transport identifier	Description
DAPARNTK	Do not supply a value on the DAPARNTK Transport Record. If you supply

Transport identifier	Description
	a value, CA Telon ignores it.
RECTYPE	TDF—Selection type.
CALCKEY	SP—CALCKEY/calckey.
CALCKYHV	SP—CALCKEY/KEY.
WHERE	SP—WHR/cond.

PROGRAM/DAREFOBJ/DARECORD/DAREQUEST/SQPARNTK Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SQPARNTK	D				1	00010000
WHERE2	P	60	A	N	4	00030000
KEY2	P	60	A	N	2	
IOAREA2	P	60	A	N	3	
HLDCUR	P	1	A	N	1	
FETCH	P	1	A	N	1	
OPTROWS	P	5	A	N	1	
SENCOLS2	P	60	A	N	4	
ORDERBY2	P	60	A	N	2	
GROUPBY2	P	60	A	N	1	
UPDCOLS	P	60	A	N	2	

Transport identifier	Description
SQPARNTK	TDF—SQL Field Extension
WHERE2	SP—DATA ACCESS/WHERE2. You can supply as many as four WHERE2 Transport records
KEY2	SP—DATA ACCESS/KEYCOLS. You can supply as many as two KEY2 Transport records
IOAREA2	SP—DATA ACCESS/IOAREA. You can supply as many as three IOAREA2 Transport records

Transport identifier	Description
HLDCUR	SP—DATA ACCESS/HLDCUR.
FETCH	SP—DATA ACCESS/FETCH.
OPTROWS	SP—DATA ACCESS/OPTROWS.
SENCOLS2	SP—DATA ACCESS/SENCOLS2. You can supply as many as four SENCOLS2 Transport records
ORDERBY2	SP—DATA ACCESS/ORDERBY. You can supply as many as two ORDERBY2 Transport records
GROUPBY2	SP—DATA ACCESS/GROUPBY.
UPDCOLS	SP—DATA ACCESS/UPDCOLS. You can supply as many as two UPDCOLS Transport records

PROGRAM/DAREFOBJ/DARECORD/DAREQEST/SQPARNT1 Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SQPARNT1	D				1	
JOINOPT	P	1	A	N	1	
ALTCURS	P	17	A	N	1	
FTCH0RI	P	1	A	N	1	
FTCHNRW	P	32	A	N	1	
FTCHNRWN	P	32	A	N	1	
FTCHSEN	P	1	A	N	1	
ISOLATE	P	1	A	N	1	

Transport identifier	Description
SQPARNT1	DB2 Fetch Details
JOINOPT	SP—Data Access with JOIN/JOINOPT
ALTCURS	SP—Data access with cursor/ALTCURS
FTCHORI	SP—Data access with cursor/FTCHORI

Transport identifier	Description
FTCHNRW	SP—Data access with cursor/FTCHNRW
FTCHRWN	SP—Data access with cursor/FTCHRWN
FTCHSEN	SP—Data access with cursor/FTCHSEN
ISOLATE	SP—Data access/ISOLATE

PROGRAM/DAREFOBJ/DARECORD/DAREQEST/SQPARNT2 Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SQPARNT2	D				1	
CASTCOL1	P	60	A	N	4	
CASTCOL2	P	60	A	N	4	
CASTAS1	P	60	A	N	4	
CASTAS2	P	60	A	N	4	

Transport identifier	Description
SQPARNT2	DB2 User Datatypes
CASTCOL1	SP—Data access/CASTCOL1. You can supply as many as four CASTCOL1 Transport records.
CASTCOL2	SP—Data access/CASTCOL2. You can supply as many as four CASTCOL2 Transport records.
CASTAS1	SP—Data access/CASTAS1. You can supply as many as four CASTAS1 Transport records.
CASTAS2	SP—Data access/CASTAS2. You can supply as many as four CASTAS2 Transport records.

PROGRAM/DAREFOBJ/DARECORD/DAREQEST/SQPARNT3 Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SQPARNT3	D				1	
NBR	P	20	N	Y	1	
TYPE	P	4	A	Y	1	
LOCFLAG	P	1	A	N	1	
CONDPTR	P	32	A	N	1	
SVCODE	P	1	A	N	1	
DIAGS	P	60	A	N	4	
GDCUST	P	8	A	N	1	
GDOPNC	P	8	A	N	1	
GDCLSC	P	8	A	N	1	

Transport identifier	Description
SQPARNT3	DB2 Get Diagnostics
NBR	SP—Data access/NBR
TYPE	SP—Data access/TYPE
LOCFLAG	SP—Data access/LOCFLAG
CONDPTR	SP—Data access/CONDPTR
SVCODE	SP—Data access/SVCODE
DIAGS	SP—Data access/DIAGS. You can supply as many as four DIAGS Transport records.
GDCUST	SP—Data access/GDCUST
GDOPNC	SP—Data access/GDOPNC
GDCLSC	SP—Data access/GDCLSC

PROGRAM/PGMENV Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
PGMENV	D	10	A	Y	1	
LINKOPT	P	1	A	N	1	
CONVERS	P	1	A	N	1	
LINEOPT	P	1	A	N	1	
PCBCMPT	P	1	A	N	1	
GENPCBS	P	1	A	N	1	
TRACE	P	1	A	N	1	
PGMNAME	P	8	A	N	1	NAME
PSBNAME	P	8	A	N	1	NAME
TRNCODE	P	10	A	N	1	TRANCD
TRANMFS	P	1	A	N	1	
TRANFLD	P	8	A	N	1	NAME
SPACMPT	P	1	A	N	1	
WKSPASZ	P	5	N	N	1	NUMERIC
WKSPAIOG	P	8	A	N	1	NAME
WKSPAIOP	P	8	A	N	1	NAME
WKSPAIN	P	1	A	N	1	
LINKPGM	P	60	A	N	5	
MSGPGM	P	60	A	N	5	
MSGTRAN	P	60	A	N	5	
MSGTBLC	P	8	A	N	1	NAME
MSGCNT	P	3	N	N	1	NUMERIC
MSGBUF	P	8	A	N	1	NAME
BUFLTH	P	4	N	N	1	NUMERIC
A4EPGM	P	6	A	N	1	NEXTPGM
A4EMSG	P	40	A	N	1	
MFSMOD	P	8	A	N	1	NAME

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SYSMSG	P	8	A	N	1	NAME
TYPE1	P	10	A	N	1	NAME
TYPE2	P	10	A	N	1	NAME
TYPE3	P	10	A	N	1	NAME
FEAT1	P	20	A	N	1	NAME
FEAT2	P	20	A	N	1	NAME
FEAT3	P	20	A	N	1	NAME
EATTR1	P	1	A	N	1	
EATTR2	P	1	A	N	1	
EATTR3	P	1	A	N	1	
USERHV	P	18	A	N	1	NAME
PASSWRD	P	18	A	N	1	NAME
PASSHV	P	18	A	N	1	NAME
SQLCON	P	1	A	N	1	YESNO
SERVER	P	1	A	N	1	YESNO
SEGEXITN	P	3	N	N	1	NUMERIC
SEGEXITV	P	3	N	N	1	NUMERIC
FRSTMOD	P	8	A	N	1	NAME
LINKDYN	P	1	A	N	1	
IOASIZE	P	5	N	N	1	NUMERIC
TPISIZE	P	5	N	N	1	NUMERIC
TPOSIZE	P	5	N	N	1	NUMERIC
USGCOPY	P	8	A	N	1	NAME
LNKCOPY	P	8	A	N	1	NAME
PSBSCHD	P	1	A	N	1	
USGCOP1	P	8	A	N	1	NAME
USGCOP2	P	8	A	N	1	NAME
BMS	P	1	A	N	1	
BMSMAP	P	8	A	N	1	NAME

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SPASTG	P	1	A	N	1	
IOASTG	P	1	A	N	1	
TPBSTG	P	1	A	N	1	
DBMS	P	1	A	N	1	
DLIWGHT	P	1	A	N	1	
STRUCTRE	P	1	A	N	1	
MIDONLY	D	1	A	N	16	
CANCEL	P	1	A	N	1	YESNO
TMF	P	1	A	N	1	YESNO

Transport identifier	Description
PGMENV	Identifies the environment where you use the parameters. Values are: ■ 'IMSPARMS' ■ 'TSOPARMS' ■ 'CICSPARMS' ' ■ BATCHPARMS' You can supply only one PGMENV Dependent Object for each value identified above. If you duplicate a value, the duplicated value is displayed in the Duplicate Key Error Message. Different Transport Records within the PGMENV Dependent Object are valid depending on the type of PGMENV you supply. The following chart shows the types of PGMENV valid for different Transport Records: ■ TSOPARMS—TSOPGM Macros (Statements/Parameters) ■ IMSPARMS—IMSPGM, IMSDRV, IMSPSB, IMSMFS, and DEVICE Macros ■ CICSPARMS—CICSPGM Macros ■ BATCHPARMS—BATCHPGM Macros Transport Records with IMSPGM Macros associated with them are only valid for ONLINE programs. Transport Records with IMSDRV Macros associated with them are only valid for DRIVER programs.
LINKOPT	SP—IMSPGM/LINKOPT
CONVERS	■ SP—IMSPGM/CONVERS

Transport identifier	Description
	■ SP—IMSDRV/CONVERS
LINEOPT	■ SP—IMSPGM/LINEOPT ■ SP—IMSMFS/LINEOPT ■ SP—CICSPGM/LINEOPT ■ SP—CICSBMS/LINEOPT
PCBCMPT	■ SP—IMSPGM/PCBCMPT ■ SP—IMSDRV/PCBCMPT ■ SP—TSOPGM/PCBCMPT
GENPCBS	■ SP—IMSPGM/GENPCBS ■ SP—IMSDRV/GENPCBS ■ SP—TSOPGM/GENPCBS ■ SP—CICSPGM/GENPCBS ■ SP—BATCHPGM/GENPCBS
TRACE	■ SP—IMSPGM/TRACE ■ SP—IMSDRV/TRACE ■ SP—CICSPGM/TRACE ■ SP—BATCHPGM/TRACE
PGMNAME	■ SP—IMSPGM/PGMNAME ■ SP—IMSDRV/PGMNAME ■ SP—BATCHPGM/PGMNAME
PSBNAME	■ SP—IMSPSB/PSBNAME ■ SP—CICSPGM/PSBNAME ■ SP—DLIPSB/PSBNAME
TRNCODE	■ SP—IMSPGM/TRANCDE ■ SP—IMSDRV/TRANCDE ■ SP—IMSMFS/TRANCDE ■ SP—CICSPGM/TRANCDE
TRANMFS	SP—IMSMFS/TRANMFS
TRANFLD	■ SP—IMSPGM/TRANFLD ■ SP—IMSMFS/TRANFLD
SPACMPT	SP—IMSPGM/SPACMPT

Transport identifier	Description
WKSPASZ	■ SP—IMSPGM/WKSPASZ ■ SP—IMSDRV/WKSPASZ The default value for ONLINE and DRIVER Programs is zero.
WKSPAIOG	■ SP—IMSPGM/WKSPAIO(GET WORKSPA MEMBER) ■ SP—IMSDRV/WKSPAIO(GET WORKSPA MEMBER)
WKSPAIOP	■ SP—IMSPGM/WKSPAIO(PUT WORKSPA MEMBER) ■ SP—IMSDRV/WKSPAIO(PUT WORKSPA MEMBER)
WKSPAIN	■ SP—IMSPGM/WKSPAIN ■ SP—IMSDRV/WKSPAIN
LINKPGM	■ SP—IMSPGM/LINKPGM ■ SP—IMSDRV/LINKPGM You can supply as many as five LINKPGM Transport Records. The System uses the entire 60 bytes of the first four Transport Records. However, the System uses only the first 16 bytes of the fifth LINKPGM record.
MSGPGM	■ SP—IMSPGM/MSGPGM ■ SP—IMSDRV/MSGPGM You can supply as many as five MSGPGM Transport Records. The System uses the entire 60 bytes of the first four Transport Records. However, the System uses only the first 16 bytes of the fifth MSGPGM record.
MSGTRAN	■ SP—IMSPGM/MSGTRAN ■ SP—IMSDRV/MSGTRAN You can supply as many as five MSGTRAN Transport Records. The System uses the entire 60 bytes of the first four Transport Records. However, the System uses only the first 16 bytes of the fifth MSGTRAN record.
MSGTBLC	■ SP—IMSPGM/MSGTBL(TABLE—MEMBER—NAME) ■ SP—IMSDRV/MSGTBL(TABLE—MEMBER—NAME)
MSGCNT	■ SP—IMSPGM/MSGTBL(NUMBER—OF—ENTRIES) ■ SP—IMSDRV/MSGTBL(NUMBER—OF—ENTRIES)
MSGBUF	■ SP—IMSPGM/MSGBUF(MEMBER—NAME) ■ SP—IMSDRV/MSGBUF(MEMBER—NAME)
BUFLTH	■ SP—IMSPGM/MSGBUF(LENGTH) ■ SP—IMSDRV/MSGBUF(LENGTH)

Transport identifier	Description
A4EPGM	■ SP—IMSPGM/A4EPGM ■ SP—IMSDRV/A4EPGM
A4EMSG	■ SP—IMSPGM/A4EMSG ■ SP—IMSDRV/A4EMSG
MFSMOD	■ SP—IMSPGM/MFSMOD ■ SP—IMSMFS/MFSMOD
SYSMSG	SP—IMSMFS/SYSMSG
TYPE1	SP—DEVICE/TYPE
TYPE2	SP—DEVICE/TYPE
TYPE3	SP—DEVICE/TYPE
FEAT1	SP—DEVICE/FEAT
FEAT2	SP—DEVICE/FEAT
FEAT3	SP—DEVICE/FEAT
EATTR1	SP—DEVICE/EATTR
EATTR2	SP—DEVICE/EATTR
EATTR3	SP—DEVICE/EATTR
SEGEXITN	SP—IMSMFS/SEGEXIT(EXITNUM)
SEGEXITV	SP—IMSMFS/SEGEXIT(EXITVECT)
FRSTMOD	SP—IMSDRV/FRSTMOD
LINKDYN	SP—IMSDRV/LINKDYN
IOASIZE	SP—IMSDRV/IOASIZE
TPISIZE	SP—IMSDRV/TPISIZE
TPOSIZE	SP—IMSDRV/TPOSIZE
USGCOPY	■ SP—IMSPGM/USGCOPY ■ SP—IMSDRV/USGCOPY ■ SP—TSOPGM/USGCOPY ■ SP—BATCHPGM/USGCOPY
LNKCOPY	■ SP—IMSPGM/LNKCOPY ■ SP—IMSDRV/LNKCOPY ■ SP—TSOPGM/LNKCOPY ■ SP—CICSPGM/LNKCOPY

Transport identifier	Description
	■ SP—BATCHPGM/LNKCOPY
PSBSCHD	SP—CICSPGM/PSBSCHD
USGCOP1	SP—CICSPGM/USGCOPY(BLL—POINTER—LIST MBRNAME)
USGCOP2	SP—CICSPGM/USGCOPY(Q—100—CICS—INIT MBRNAME)
BMS	SP—CICSPGM/BMS
BMSMAP	■ SP—CICSPGM/BMSMAP ■ SP—CICSBMS/BMSMAP
SPASTG	SP—CICSPGM/SPASTG
IOASTG	SP—CICSPGM/IOASTG
TPBSTG	SP—CICSPGM/TPBSTG
DBMS	■ TDF &.—UPDATE CICS ENV SCREEN/DBMS ■ TDF—UPDATE BATCH ENV/DBMS ■ SP—DLIDSC/DBMS
DLIWGHT	SP—BATCHPGM/DLIWGHT
STRUCTRE	SP—BATCH/STRUCTRE
MIDONLY	MFS MID Only field definitions. You can supply as many as 16 MIDONLY definitions. If you exceed this limit, the Key_Value in the Duplicate Key Error Message contains 'LIMIT REACHED.'
CANCEL	Use this field to activate logic in the Driver Program to delete the program from memory after the CALLed program has returned control. Values are: ■ N ■ Y

PROGRAM/PGMENV/MIDONLY Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
MIDONLY	D	1	A	Y	1	
MIDSRC	P	8	A	N	1	NAME
LABEL	P	8	A	N	1	NAME

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
INIT	P	40	A	N	1	INIT
LTH	P	3	N	N	1	NUMERIC

Transport identifier	Description
MIDONLY	Do not supply a value on the MIDONLY Transport Record. If you supply a value, it is ignored.
MIDSRC	TDF—UPDATE IMS MFS/MIDSOURCE SP—FIELD/FMTCNTL
LABEL	TDF—UPDATE IMS MFS/LABEL SP—FIELD/LABEL
INIT	TDF—UPDATE IMS MFS/INITIALIZATION SP—FIELD/INIT
LTH	TDF—UPDATE IMS MFS/LTH SP—FIELD/LTH

PROGRAM/SORT Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SORT	D	8	A	Y	1	NAME
DESC	P	40	A	N	1	
INFILE	P	8	A	N	1	NAME
OUTFILE	P	8	A	N	1	NAME
COLLATE	P	8	A	N	1	
STORAGE	P	7	N	N	1	NUMERIC
INPROC	P	8	A	N	1	NAME
OUTPROC	P	8	A	N	1	NAME
LRECL1	P	4	N	N	1	NUMERIC
LRECL2	P	4	N	N	1	NUMERIC

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
COPY	P	8	A	N	1	NAME
COPYLV1	P	1	A	N	1	
COPYLBL	P	30	A	N	1	NAME
PREFIX	P	4	A	N	1	
SORTKEY	D	1	A	Y	12	

Transport identifier	Description
SORT	SP—SORT/LABEL.
DESC	SP—SORT/DESCRIPTION.
INFILE	SP—SORT/INFILE.
OUTFILE	SP—SORT/OUTFILE.
COLLATE	SP—SORT/COLLATE.
STORAGE	SP—SORT/STORAGE.
INPROC	SP—SORT/INPROC.
OUTPROC	SP—SORT/OUTPROC.
LRECL1	SP—SORT/LRECL1.
LRECL2	SP—SORT/LRECL2.
COPY	SP—SORT/COPY.
COPYLVL1	SP—SORT/COPYLVL1.
COPYLBL	SP—SORT/COPYLBL.
PREFIX	SP—SORT/PREFIX.
SORTKEY	Sort Key Definitions. You can supply as many as 12 SORTKEY definitions can be supplied. If you exceed this limit, the Key_Value in the Duplicate Key Error Message contains 'LIMIT REACHED'.

PROGRAM/SORT/SORTKEY Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SORTKEY	D	1	A	Y	1	
START	P	4	N	Y	1	
LTH	P	3	N	Y	1	NUMERIC
SIZE1	P	2	N	N	1	NUMERIC
SIZE2	P	2	N	N	1	NUMERIC
ORDER	P	1	A	N	1	
FORM	P	2	A	N	1	
DATANAME	P	60	A	N	1	OFNAME

Transport identifier	Description
SORTKEY	SP—SORTKEY/SORTKEY
START	SP—SORTKEY/START
LTH	SP—SORTKEY/LTH
SIZE1	SP—SORTKEY/SIZE1
SIZE2	SP—SORTKEY/SIZE2
ORDER	SP—SORTKEY/ORDER
FORM	SP—SORTKEY/FORM
DATANAME	SP—SORTKEY/DATANAME

PROGRAM/MERGE Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
MERGE	D	1	A	Y	1	
MAXLRECL	P	5	N	N	1	NUMERIC
MERGEFIL	P	8	A	Y	20	

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
MERGEGRP	D	2	A	Y	19	

Transport identifier	Description
MERGE	
MAXLRECL	SP—MERGE/MAXLRECL.
MERGEFIL	SP—MERGE/MERGE.
MERGEGRP	Merge Group Definitions. You can supply as many as 18 MERGEGRP definitions. If you exceed this limit, the Key_Value in the Duplicate Key Error Message contains 'LIMIT REACHED'.

PROGRAM/MERGE/MERGEGRP Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
MERGEGRP	D	2	A	Y	1	
DESC	P	30	A	N	1	
FORM	P	1	A	N	1	
LTH	P	4	N	N	1	NUMERIC
SIZE1	P	5	N	N	1	NUMERIC
SIZE2	P	5	N	N	1	NUMERIC
MERGEKEY	D	1	A	Y	20	

Transport identifier	Description
MERGEGRP	SP—MERGEGRP/GROUPNUM.
DESC	SP—MERGEGRP/DESC.
FORM	SP—MERGEGRP/FORM.
LTH	SP—MERGEGRP/LTH.
SIZE1	SP—MERGEGRP/SIZE1.
SIZE2	SP—MERGEGRP/SIZE2.

Transport identifier	Description
MERGEKEY	Merge Key Definitions. You can supply as many as 20 MERGEKEY definitions. If you exceed this limit, the Key_Value in the Duplicate Key Error Message contains 'LIMIT REACHED'.

PROGRAM/MERGE/MERGEGRP/MERGEKEY Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
MERGEKEY	D	2	A	Y	1	
DATANAME	P	60	A	Y	1	OFNAME
FILENAME	P	8	A	Y	1	NAME
LABEL	P	8	A	Y	1	NAME

Transport identifier	Description
MERGEKEY	SP—MERGEKEY/ORDER
DATANAME	SP—MERGEKEY/DATANAME
FILENAME	SP—MERGEKEY/FILENAME

PROGRAM/SPPARAM Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SPPARAM	D	18	A	Y	1	NAME
KIND	P	1	A	Y	1	
LTH	P	5	N	Y	1	NUMERIC
TYPE	P	1	A	Y	1	
DEC	P	3	N	N	1	NUMERIC
IND	P	1	A	N	1	
DBNAME	P	60	A	N	1	NAME

Transport identifier	Description
SPPARAM	Supply name of parameter
KIND	SP—PARAM/KIND
LTH	SP—PARAM/LTH
TYPE	SP—PARAM/TYPE
DEC	SP—PARAM/DEC
IND	SP—PARAM/IND
DBNAME	SP—PARAM/DBNAME

PROGRAM/STPROCKY Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
STPROCKY	D	1	A	Y	1	
PRESP	P	8	A	N	1	NAME
POSTSP	P	8	A	N	1	NAME
SPHDR	P	5	A	Y	1	NAME
SPID	P	5	A	Y	1	NAME
IGNORE	P	30	A	N	1	LIST

Transport identifier	Description
STPROCKY	No value needed. Value ignored.
PRESP	SP—STPROC/PRESP
POSTSP	SP—STPROC/POSTSP
SPHDR	SP—STPROC/SPHDR
SPID	SP—STPROC/SPID
IGNORE	SP—STPROC/IGNORE

PROGRAM/MATCHKEY

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
MATCHKEY	D	1	A	Y	1	
MASTKEY	P	60	A	Y	1	
TRANKEY	P	60	A	Y	1	
MLENGTH	P	3	N		1	
MSIZE1	P	2	N		1	
MSIZE2	P	2	N		1	
MFORM	P	1	A		1	

Transport identifier	Description
MATCHKEY	
MASTKEY	SP—MATCH/MASTKEY
TRANKEY	SP—MATCH/TRANKEY
MLENGTH	SP—MATCH/MLENGTH
MSIZE1	SP—MATCH/MSIZE1
MSIZE2	SP—MATCH/MSIZE2
MFORM	SP—MATCH/MFORM

PROGRAM/LANGENV

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
LANGENV	D	10	A	Y	1	
ALIGN	P	1	A	N	1	
STORAGE	P	1	A	N	1	
REORDER	P	1	A	N	1	
XOPTS	P	60	A	N	1	

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
IGNORE	P	30	A	N		

Transport identifier	Description
LANGENV	SP—PLIXOPT/environment. Values are: ■ 'PLIXOPT—IMS' ■ 'PLIXOPT—TSO' ■ 'PLIXOPT—CICS' ■ 'PLIXOPT—BATCH' ■ 'PLIXOPT—ALL' You can supply only one LANGENV Dependent Object for each value identified above. If you duplicate a value, the duplicated value is displayed in the Duplicate Key Error Message.
ALIGN	SP—PLIXOPT/ALIGN.
STORAGE	SP—PLIXOPT/STORAGE.
REORDER	SP—PLIXOPT/REORDER.
XOPTS	SP—PLIXOPT/XOPTS.
IGNORE	SP—STPROC/IGNORE

PROGRAM/CUSTCODE Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
CUSTCODE	D	8	A	Y	1	NBALPHA
TYPE	P	5	A	Y	1	
JCL	P	3	A	N	1	
DESC	P	40	A	N	1	
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
CRTDATE	P	8	N	N	1	NUMERIC
LASTUSED	P	8	N	N	1	NUMERIC
BEGIN	P	8	A	Y	1	
CARDIMAG	D	80	A	N	9999	

Transport identifier	Description
CUSTCODE	■ TDF—ONLINE PROGRAM DEFINITION MENU/CUSTCODE ■ TDF—ONLINE BATCH DEFINITION MENU/CUSTCODE Identifies the name of the Custom Code Member.
TYPE	■ SP—SCREEN/LANG ■ SP—BATCH/LANG ■ SP—REPORT/LANG ■ SP—DRIVER/LANG Values are: ■ 'COBOL' ■ 'PLI'
JCL	Identifies the Custom Code Member as containing JCL Statements instead of COBOL or PL/I Source Code. This allows you to submit the Custom Code Member to the Internal Reader in the Custom Code Editor. Values are: ■ YES—Contains JCL ■ NO—Contains COBOL or PL/I code
DESC	■ TDF—ONLINE PROGRAM DEFINITION MENU/DESC ■ TDF—BATCH PROGRAM DEFINITION MENU/DESC
LASTUSER	TDF—CUSTOM CODE EDITOR/LAST USER TO UPDATE.
LASTUPD	TDF—CUSTOM CODE EDITOR/LAST UPDATE DATE.
UPDTIME	TDF—CUSTOM CODE EDITOR/LAST UPDATE TIME.
CRTDATE	TDF—CUSTOM CODE EDITOR/CREATE DATE.
LASTUSED	TDF—CUSTOM CODE EDITOR/DATE LAST ACCESSED.
BEGIN	Identifies the start of the actual Custom Code. The Data Value must contain the value 'CUSTCODE'.

Transport identifier	Description
CARDIMAG	COBOL, PL/I or JCL associated with the Custom Code Member. You can supply as many as 9,999 lines of code for a member. When you exceed this limit, the Key_Value in the Duplicate Key Error Message contains 'LIMIT REACHED'.

PROGRAM/CUSTCODE/CARDIMAG Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
CARDIMAG	D	80	A	Y	1	

Transport identifier	Description
CARDIMAG	Line of Custom Code. The CARDIMAG Transport Record is unformatted and can contain any value. Indicate the end of the CARDIMAG Transport Records by supplying the END CUSTCODE Transport Record.

IMSDBD Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
IMSDBD	R	8	A	Y	1	
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	N	N	1	NUMERIC
CRTDATE	P	8	N	N	1	NUMERIC
LASTUSED	P	8	N	N	1	NUMERIC
DESC	P	40	A	N	1	
LNAMES	P	60	A	N	3	
ACCESS	P	28	A	N	1	
RMNAME	P	28	A	N	1	

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
PASSWD	P	3	A	N	1	
DATA SET	D	8	A	N	99999	
AREA	D	8	A	N	-	
SEGM	D	8	A	N	-	
LCHILD	D	8	A	N	-	

Transport identifier	Description
IMSDBD	■ SP—DBD/DBDNAME1 ■ TDF—DATA ADMINISTRATION MENU/NAME
LASTUSER	TDF—IMS DBD/LAST USER TO UPDATE.
LASTUPD	TDF—IMS DBD/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—IMS DBD/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—IMS DBD/DATE CREATED (YYMMDD).
LASTUSED	TDF—IMS DBD/LAST ACCESSED (YYMMDD).
DESC	TDF—DATA ADMINISTRATION MENU/DESC.
LNAMES	SP—DBD/DBDNAME2,,,,n. The LNAMES Transport Record allows the specification of any additional logical DBD names for a DBD. The System performs no editing on the values(s) supplied. You can supply as many as three LNAMES Transport Records. The System uses only the first 24 bytes of the third Transport Record.
ACCESS	■ SP—DBD/ACCESS ■ TDF—CREATE/UPDATE DBD/ACCESS This record allows the specification of the access methods used for a DBD (for example, HIDAM and VSAM). The system performs no editing on the value supplied.
RMNAME	■ SP—DBD/RMNAME ■ TDF—CREATE/UPDATE DBD/RMNAME This record allows the specification of the randomizing routines and parameters used for a DBD. The system performs no editing on the value supplied.
PASSWD	SP—DBD/PASSWD.
DATA SET	Identifies DATA SETs that are defined for the DBD. You can supply DATA

Transport identifier	Description
	SET, AREA, SEGM, and LCHILDs in any sequence. You can supply as many as 99,999 of combined DATA SET, AREA, SEGM, or LCHILD Dependent Objects for an IMSDBD. If you exceed this limit, the Key_Value in the Duplicate Key Error Message is replaced with the value 'LIMIT REACHED'.
AREA	Identifies AREAs that are defined for the DBD. AREAs are only valid for Fast Path DBDs.
SEGM	Identifies SEGMs that are defined for the DBD.
LCHILD	Identifies LCHILDSs that are defined for the DBD.

IMSDBD/DATA SET Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
DATA SET	D	8	A	Y	1	NAME
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
DEVICE	P	4	A	N	1	NAME
MODEL	P	2	A	N	1	
OVFLW	P	8	A	N	1	NAME
BLOCK	P	5	A	N	1	
SIZE	P	5	A	N	1	
RECORD	P	13	A	N	1	
RECFM	P	2	A	N	1	
SCAN	P	3	A	N	1	
FRSPC	P	8	A	N	1	
REL	P	18	A	N	1	

Transport identifier	Description
DATA SET	■ SP—DATA SET/DD1 ■ TDF—CREATE/UPDATE DBD/NAME
LASTUSER	TDF—DATA SET/LAST USER TO UPDATE.
LASTUPD	TDF—DATA SET/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—DATA SET/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—DATA SET/DATE CREATED (YYMMDD).
DEVICE	SP—DATA SET/DEVICE.
MODEL	SP—DATA SET/MODEL.
OVFLW	SP—DATA SET/OVFLW.
BLOCK	SP—DATA SET/BLOCK. This record allows the specification of the BLKFACT1 and BLKFACT2 values or the SIZE0 values for a DATA SET. The System performs no editing on the value supplied.
SIZE	SP—DATA SET/SIZE. This record allows the specification of the SIZE1 and SIZE2 values for a DATA SET. The System performs no editing on the value supplied.
RECORD	SP—DATA SET/RECORD. This record allows the specification of the RECLEN1 and RECLEN2 values for a DATA SET. The System performs no editing on the value supplied.
RECFM	SP—DATA SET/RECFM.
SCAN	SP—DATA SET/SCAN.
FRSPC	SP—DATA SET/FRSPC. This record allows the specification of the FBFF and FSPF values for a DATA SET. The System performs no editing on the value supplied.
REL	SP—DATA SET/REL. This record allows the specification of the REL values for a DATA SET. The System performs no editing on the value supplied.

IMSDBD/AREA Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
AREA	D	8	A	Y	1	NAME
LASTUSER	P	8	A	N	1	

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
DEVICE	P	4	A	N	1	
MODEL	P	2	A	N	1	
SIZE	P	5	A	N	1	
UOW	P	13	A	N	1	
ROOT	P	13	A	N	1	

Transport identifier	Description
AREA	■ SP—AREA/DD1 ■ TDF—CREATE/UPDATE DBD/NAME
LASTUSER	TDF—AREA/LAST USER TO UPDATE.
LASTUPD	TDF—AREA/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—AREA/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—AREA/DATE CREATED (YYMMDD).
DEVICE	SP—AREA/DEVICE.
MODEL	SP—AREA/MODEL.
SIZE	SP—AREA/SIZE. This record allows the specification of the SIZE1 and SIZE2 values for an AREA. The System performs no editing on the value supplied.
UOW	SP—AREA/UOW. This record allows the specification of the NUMBER1 and OVERFLOW1 values for an AREA. The System performs no editing on the value supplied.
ROOT	SP—AREA/ROOT. This record allows the specification of the NUMBER2 and OVERFLOW2 values for an AREA. The System performs no editing on the value supplied.

IMSDBD/SEGM Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SEGM	D	8	A	Y	1	NAME
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
PARENT	P	8	A	Y	1	
PPOINTER	P	4	A	N	1	
LPARENT	P	8	A	N	1	NAME
VIRTUAL	P	8	A	N	1	
PDBNAME	P	8	A	N	1	NAME
MINBYTES	P	5	N	Y	1	NUMERIC
MAXBYTES	P	5	N	Y	1	NUMERIC
TYPE	P	3	A	N	1	
FREQ	P	12	A	N	1	
POINTER	P	36	A	N	1	
RULES	P	11	A	N	1	
COMPRTN	P	20	A	N	1	
SRCESEG1	P	8	A	N	1	NAME
SRCETYP1	P	4	A	N	1	
SRCEDBN1	P	8	A	N	1	NAME
SRCESEG2	P	8	A	N	1	NAME
SRCETYP2	P	4	A	N	1	
SRCEDBN2	P	8	A	N	1	NAME
KEYFLD	P	8	A	N	1	NAME
KEYSEQ	P	3	A	N	1	
UNIQMULT	P	8	A	N	1	
KEYBYTES	P	5	N	N	1	NUMERIC

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
KEYSTART	P	5	N	N	1	NUMERIC
KEYTYPE	P	1	A	N	1	
LABEL	P	8	A	N	1	NAME
COPY	P	8	A	N	1	NAME
COPYLV1	P	1	A	N	1	
COPYLBL	P	30	A	N	1	NAME
KEYPIC	P	23	A	N	1	
OPCODE	P	5	A	N	1	
KEY	P	60	A	N	1	OFNAME
CMDCODE	P	4	A	N	1	
PROCOPT	P	5	A	N	1	
INDICES	P	60	A	N	1	
DLIDSC	D	8	A	N	—	
SRCHFLD	D	8	A	N	99999	

Transport identifier	Description
SEGM	SP—SEGM/NAME.
LASTUSER	TDF—SEGM/LAST USER TO UPDATE.
LASTUPD	TDF—SEGM/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—SEGM/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—SEGM/DATE CREATED (YYMMDD).
PARENT	■ SP—SEGM/PARENT(SEGNAME2) ■ TDF—CREATE/UPDATE DBD/PARENT
PPOINTER	SP—SEGM/PARENT(SNGL,DBLE).
LPARENT	SP—SEGM/PARENT(LPSEGNAME).
VIRTUAL	SP—SEGM/PARENT(VIRTUAL,PHYSICAL). Values are: ■ 'VIRTUAL' ■ 'PHYSICAL' ■ 'DATA'.

Transport identifier	Description
PPDBNAME	SP—SEGM/PARENT(DBNAME1).
MINBYTES	SP—SEGM/BYTES(MIN BYTES).
MAXBYTES	■ SP—SEGM/BYTES(MAX BYTES) ■ SP—DLIDSC/SEGLTH ■ TDF—CREATE/UPDATE DBD/SEGLTH
TYPE	SP—SEGM/TYPE.
FREQ	SP—SEGM/FREQ.
POINTER	SP—SEGM/POINTER. This record allows the specification of the POINTER values for a SEGM. The system performs no editing on the value supplied.
RULES	SP—SEGM/RULE. This record allows the specification of the RULES values for a SEGM. The system performs no editing on the value supplied.
COMPRTN	SP—SEGM/COMPRTN. This record allows the specification of the COMPRTN values for a SEGM. The system performs no editing on the value supplied.
SRCESEG1	SP—SEGM/SOURCE(SEGNAME).
SRCETYP1	SP—SEGM/SOURCE(KEY,DATA).
SRCEDBN1	SP—SEGM/SOURCE(DBNAME).
SRCESEG2	SP—SEGM/SOURCE(SEGNAME).
SRCETYP2	SP—SEGM/SOURCE(KEY,DATA).
SRCEDBN2	SP—SEGM/SOURCE(DBNAME).
KEYFLD	SP—SEGMENT/IMSKEY. Identifies the name of the key FIELD for a SEGM. The KEYSEQ, UNIQMULT, KEYBYTES, KEYSTART, and KEYTYPE Transport Records specify the FIELD macro parameters for the SEGM's key field. This information should also be duplicated in a SRCHFLD Transport Dependent Object.
KEYSEQ	SP—FIELD/NAME(SEQ). The FIELD macro referenced above is the IMS DBD FIELD Macro, not the macro used in the PANEL Root Object.
UNIQMULT	SP—FIELD/NAME(U,M). Values are: ■ MULTIPLE ■ UNIQUE The FIELD macro referenced above is the IMS DBD FIELD macro, not the macro used in the PANEL Root Object.
KEYBYTES	■ SP—FIELD/BYTES ■ SP—SEGMENT/KEYLEN

Transport identifier	Description
	■ SP—DLIDSC/KEYLTH The FIELD macro referenced above is the IMS DBD FIELD macro, not the one macro used in the PANEL Root Object. On the DLIDSC macro the KEYBYTES parameter contains the concatenated key lengths of all SEGMs in the hierarchical path.
KEYSTART	SP—FIELD/START. The FIELD Macro referenced above is the IMS DBD FIELD Macro, not the macro used in the PANEL Root Object.
KEYTYPE	SP—FIELD/TYPE. The FIELD Macro referenced above is the IMS DBD FIELD Macro, not the macro used in the PANEL Root Object.
LABEL	SP—SEGMENT/LABEL.
COPY	SP—SEGMENT/COPY.
COPYLV1	SP—SEGMENT/COPYLV1.
COPYLBL	SP—SEGMENT/COPYLBL.
KEYPIC	■ SP—SEGMENT/KEYPIC ■ SP—DLIDSC/KEYPIC
OPCODE	SP—SEGMENT/OPCODE.
KEY	SP—SEGMENT/SEGKEY.
CMDCODE	SP—SEGMENT/CMDCODE.
PROCOPT	SP—SEGMENT/PROCOPT.
INDICES	SP—SEGMENT/INDICES.
DLIDSC	DL/I Data Search Criteria associated with a SEGM.
SRCHFLD	FIELDs contained within a SEGM. You can supply as many as 99,999 SRCHFLDs for a SEGM. If you exceed this limit, the Key_Value in the Duplicate Key Error Message is replaced with the value 'LIMIT REACHED'.

IMSDBD/SEGM/DLIDSC Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
DLIDSC	D	8	A	Y	1	NAME
LASTUSER	P	8	A	N	1	

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
IOAREA	P	60	A	N	1	OFNAME
KEYFEED	P	30	A	N	1	NAME
CMDCODE	P	4	A	N	1	
PATH	P	1	A	N	1	YESNO
CONCATK	P	1	A	N	1	YESNO
CURRENT	P	1	A	N	1	YESNO
PARENTG	P	1	A	N	1	YESNO
OPTION	P	1	A	N	1	
LOCKED	P	1	A	N	1	YESNO
VARLTH	P	1	A	N	1	YESNO
OFFSET	P	5	N	N	1	NUMERIC
PROCSEQ	P	8	A	N	1	NAME
DSCQUAL	D	1	A	N	12	

Transport identifier	Description
DLIDSC	SP—DLIDSC/LABEL. DLIDSCs are keyed by the Data Value supplied on the DLIDSC Transport Record and must be unique within a SEGM. If duplicates are found, the Key_Value in the Duplicate Key Error Message contains the name of the duplicated DSC.
LASTUSER	TDF—DLIDSC/LAST USER TO UPDATE.
LASTUPD	TDF—DLIDSC/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—DLIDSC/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—DLIDSC/DATE CREATED (YYMMDD).
IOAREA	SP—DLIDSC/IOAREA.
KEYFEED	SP—DLIDSC/KEYFEED.
CMDCODE	SP—DLIDSC/CMDCODE.
PATH	SP—DLIDSC/PATH.

Transport identifier	Description
CONCATK	SP—DLIDSC/CONCATK.
CURRENT	SP—DLIDSC/CURRENT.
PARENTG	SP—DLIDSC/PARENTG.
OPTION	SP—DLIDSC/OPTION.
LOCKED	SP—DLIDSC/LOCKED.
VARLTH	SP—DLIDSC/VARLTH.
OFFSET	SP—DLIDSC/OFFSET.
PROCSEQ	TDF—UPDATE. SSA/COMMAND FOR DL/I SEGMENT/PROCSEQ.
DSCQUAL	SSA Qualifications for a DLIDSC. You can specify as many as 12 DSCQUALs for each DLIDSC. If you exceed this limit, the Key_Value in the Duplicate Key Error Message is replaced with the value 'LIMIT REACHED'.

IMSDBD/SEGM/DLIDSC/DSCQUAL Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
DSCQUAL	D	1	A	N	1	
FLDNAME	P	8	A	N	1	NAME
OPCODE	P	2	A	N	1	
SEGKEY	P	60	A	N	1	
BOOLEAN	P	3	A	N	1	

Transport identifier	Description
DSCQUAL	SSA Qualification Value. The System supplies no value on the DSCQUAL Transport Record. If you supply a value, it is ignored.
FLDNAME	SP—DLIDSC/IMSKEY.
OPCODE	SP—DLIDSC/OPCODE.
SEGKEY	SP—DLIDSC/KEY.
BOOLEAN	SP—DLIDSC/BOOLEAN. Values are: ■ 'AND' ■ 'OR'

Transport identifier	Description
	■ '#' ■ '+' ■ '|' ■ '*'

IMSDBD/SEGM/SRCHFLD Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SRCHFLD	D	8	A	Y	1	NAME
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
KEYFLD	P	3	A	N	1	
KEYTYPE	P	8	A	N	1	
BYTES	P	5	N	Y	1	NUMERIC
START	P	5	N	Y	1	NUMERIC
TYPE	P	1	A	Y	1	

Transport identifier	Description
SRCHFLD	FIELD/NAME. **Note:** All references to the FIELD Macro in the SRCHFLD Dependent Object refer to the IMS DBD FIELD Macro and not the PANEL Root Object FIELD Macro.
LASTUSER	TDF—FIELD/LAST USER TO UPDATE.
LASTUPD	TDF—FIELD/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—FIELD/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—FIELD/DATE CREATED (YYMMDD).
KEYFLD	■ SP—FIELD/NAME(SEQ) ■ TDF—LIST SEARCH FIELDS/DARECORDKEY

Transport identifier	Description
KEYTYPE	SP—FIELD/NAME(U,M). Values are: ■ 'UNIQUE' ■ 'MULTIPLE'
BYTES	■ SP—FIELD/BYTES ■ TDF—LIST SEARCH FIELDS/SCHFLD LENGTH
START	■ SP—FIELD/START ■ TDF—LIST SEARCH FIELDS/SCHFLD START
TYPE	■ SP—FIELD/TYPE ■ TDF—LIST SEARCH FIELDS/SCHFLD TYPE

IMSDBD/LCHILD Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
LCHILD	D	8	A	Y	1	NAME
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
LDBNAME	P	8	A	N	1	NAME
POINTER	P	4	A	N	1	
PAIR	P	8	A	N	1	NAME
INDEX	P	8	A	N	1	NAME
RULES	P	5	A	N	1	
MAXBYTES	P	5	N	N	1	NUMERIC
KEYFLD	P	8	A	N	1	NAME
KEYBYTES	P	5	N	N	1	NUMERIC
KEYSTART	P	5	N	N	1	NUMERIC
XDFLD	S	8	A	N	99999	

Transport identifier	Description
LCHILD	SP—LCHILD/NAME(SEGNAME1).
LASTUSER	TDF—LCHILD/LAST USER TO UPDATE.
LASTUPD	TDF—LCHILD/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—LCHILD/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—LCHILD/DATE CREATED (YYMMDD).
LDBNAME	SP—LCHILD/NAME(DBNAME).
POINTER	SP—LCHILD/POINTER.
PAIR	SP—LCHILD/PAIR.
INDEX	SP—LCHILD/INDEX.
RULES	SP—LCHILD/RULES.
MAXBYTES	TDF—UPDATE DBD/MAX LTH.
KEYFLD	TDF—UPDATE DBD/DARECORDKEY.
KEYBYTES	TDF—UPDATE DBD/LENGTH.
KEYSTART	TDF—UPDATE DBD/START.
XDFLD	Identifies the IMS XDFLDs associated with an LCHILD. You can supply as many as 99,999 XDFLDs for an LCHILD. If you exceed this limit, the Key_Value in the Duplicate Key Error Message is replaced with the value 'LIMIT REACHED'.

IMSDBD/LCHILD/XDFLD Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
XDFLD	D	8	A	Y	1	NAME
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
SEGMENT	P	8	A	N	1	NAME
CONST	P	1	A	N	1	

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SRCH	P	40	A	N	1	
SUBSEQ	P	46	A	N	1	
DDATA	P	46	A	N	1	
NULLVAL	P	1	A	N	1	
EXTRTN	P	8	A	N	1	NAME

Transport identifier	Description
XDFLD	SP—XDFLD/NAME.
LASTUSER	TDF—XDFLD/LAST USER TO UPDATE.
LASTUPD	TDF—XDFLD/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—XDFLD/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—XDFLD/DATE CREATED (YYMMDD).
SEGMENT	SP—XDFLD/SEGMENT.
CONST	SP—XDFLD/CONST.
SRCH	SP—XDFLD/SRCH.
SUBSEQ	SP—XDFLD/SUBSEQ.
DDATA	SP—XDFLD/DDATA.
NULLVAL	SP—XDFLD/NULLVAL.
EXTRTN	SP—XDFLD/EXTRTN.

TABLE Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
TABLE	R	4	A	Y	1	
QUAL	P	8	A	Y	1	NAME
TBLNAME	P	18	A	Y	1	NAME
LASTUSER	P	8	A	N	1	

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	N	N	1	NUMERIC
CRTDATE	P	6	N	N	1	NUMERIC
DESC	P	40	A	N	1	
TLNNAME	P	12	A	Y	1	NAME
LABEL	P	8	A	N	1	NAME
DCLCOPY	P	8	A	Y	1	NAME
DCLLBL	P	30	A	N	1	NAME
DCLRDEF	P	1	A	N	1	YESNO
COPY	P	8	A	N	1	NAME
COPYLBL	P	30	A	N	1	NAME
COPYLV1	P	1	A	N	1	YESNO
SYNONYM	P	1	A	N	1	YESNO
TLNROW	D	8	A	N	99999	

Transport identifier	Description
TABLE	Identifies the type of DB2/SQL table defined. Values are: ■ 'BASE' (Default) ■ 'VIEW' TABLEs are keyed by the Qualifier/Table Name supplied in the QUAL and TBLNAME Transport Records. If another TABLE or JOIN exists under that name, the Key_Value in the Duplicate Key Error Message contains the QUAL's Transport Record value followed by the TBLNAME's Transport Record value. TABLEs are also keyed by the Label Transport Record. If another TABLE or JOIN has the same Label, then the Key_Value in the Duplicate Key Error Message contains the duplicate Label name.
QUAL	SP—TABLE/QUAL.
TBLNAME	SP—TABLE/NAME.
LASTUSER	TDF—DB2/SQL TABLE/LAST USER TO UPDATE.
LASTUPD	TDF—DB2/SQL TABLE/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—DB2/SQL TABLE/TIME LAST UPDATED (HHMM).

Transport identifier	Description
CRTDATE	TDF—DB2/SQL TABLE/DATE CREATED (YYMMDD).
DESC	SP—TABLE/DESCR.
TLNNAME	SP—TABLE/TLNNAME.
LABEL	SP—TABLE/LABEL.
DCLCOPY	SP—TABLE/DCLCOPY.
DCLLBL	SP—TABLE/DCLLBL.
DCLRDEF	SP—TABLE/DCLRDEF.
COPY	SP—TABLE/COPY.
COPYLBL	SP—TABLE/COPYLBL.
COPYLV1	SP—TABLE/COPYLV1.
SYNONYM	SP—TABLE/SYNONYM.
TLNROW	Identify the rows associated with a TABLE. You can supply as many as 99,999 TLNROWs for a TABLE. If you exceed this limit, the Key_Value in the Duplicate Key Error Message is replaced with the value 'LIMIT REACHED.'

TABLE/TLNROW Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
TLNROW	D	8	A	Y	1	NAME
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
DCLCOL	D	18	A	N	99999	
TMPTBL	P	1	A	N	1	YESNO

Transport identifier	Description
TLNROW	SP—TLNROW/LABEL.

Transport identifier	Description
LASTUSER	TDF—SQL ROW/LAST USER TO UPDATE.
LASTUPD	TDF—SQL ROW/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—SQL ROW/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—SQL ROW/DATE CREATED (YYMMDD).
DCLCOL	DB2 Column Definitions. You can supply as many as 99,999 DCLCOLs for a TLNROW. If you exceed this limit, the Key_Value in the Duplicate Key Error Message is replaced with the value 'LIMIT REACHED.'
TEMPTBL	Temporary table indicator. If set to Y, the TLNROW is treated as a temporary table when the table it belongs to is DGADDed to a program's Data Group.

TABLE/TLNROW/DCLCOL Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
DCLCOL	D	18	A	Y	1	NAME
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
ALIAS	P	30	A	N	1	NAME
KEY	P	2	A	N	1	
NOTNULL	P	1	A	N	1	YESNO
TYPE	P	8	A	Y	1	
LTH	P	4	N	Y	1	NUMERIC
DEC	P	2	N	N	1	NUMERIC
TEMPTBL	P	1	A	N	1	YESNO
ACCESS	P	1	A	N	1	YESNO

Transport identifier	Description
DCLCOL	■ SP—DCLCOL/NAME

Transport identifier	Description
	■ SP—DCLCOL/DCLHOST
LASTUSER	TDF—SQL COLUMN/LAST USER TO UPDATE
LASTUPD	TDF—SQL COLUMN/DATE LAST UPDATED (YYMMDD)
UPDTIME	TDF—SQL COLUMN/TIME LAST UPDATED (HHMM)
CRTDATE	TDF—SQL COLUMN/DATE CREATED (YYMMDD)
ALIAS	SP—DCLCOL/ALIAS
KEY	SP—DCLCOL/KEY
NOTNULL	SP—DCLCOL/NOTNULL
TYPE	SP—DCLCOL/TYPE. Values are: ■ 'INTEGER' ■ 'SMALLINT' ■ 'FLOAT' ■ 'DECIMAL' ■ 'CHAR ■ 'VARCHAR' ■ 'LONGVARC' ■ 'GRAPHIC' ■ 'VARGRAPH' ■ 'LONGVARG' ■ 'DATE' ■ 'TIME' ■ 'TIMESTMP'
LTH	SP—DCLCOL/LTH
DEC	SP—DCLCOL/DEC
TEMPTBL	SP—TLNROW/TEMPTBL
ACCESS	SP—DCLCOL/ACCESS

JOIN Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
JOIN	R	1	A	Y	1	
QUAL	P	8	A	Y	1	NAME
TBLNAME	P	18	A	Y	1	NAME
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	N	N	1	NUMERIC
CRTDATE	P	6	N	N	1	NUMERIC
DESC	P	40	A	N	1	
TLNNAME	P	12	A	Y	1	NAME
LABEL	P	8	A	N	1	NAME
DCLCOPY	P	8	A	Y	1	NAME
DCLLBL	P	30	A	N	1	NAME
DCLRDEF	P	1	A	N	1	YESNO
COPY	P	8	A	N	1	NAME
COPYLBL	P	30	A	N	1	NAME
COPYLV1	P	1	A	N	1	YESNO
SYNONYM	P	1	A	N	1	YESNO
JOINTBL	D	1	A	N	15	
JOINCOL	D	1	A	N	99999	
TLNROW	D	8	A	N	99999	

Transport identifier	Description
JOIN	Do not supply a value on the JOIN Transport Record. If you supply a value, it is ignored. JOINs are keyed by the Qualifier/Table Name supplied in the QUAL and TBLNAME Transport Records. If another JOIN or TABLE exists under that name, the Key_Value in the Duplicate Key Error Message contains the QUAL's Transport Record value followed by the TBLNAME's Transport Record value. JOINs are also keyed by the Label Transport Record. If another JOIN or TABLE has the same Label, then the Key_Value in the Duplicate Key Error Message contains the duplicate

Transport identifier	Description
	Label name.
QUAL	SP—TLNJOIN/QUAL.
TBLNAME	SP—TLNJOIN/NAME.
LASTUSER	TDF—SQL TABLE/LAST USER TO UPDATE.
LASTUPD	TDF—SQL TABLE/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—SQL TABLE/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—SQL TABLE/DATE CREATED (YYMMDD).
DESC	SP—TABLE/DESCR.
TLNNAME	SP—TABLE/TLNNAME.
LABEL	SP—TLNJOIN/LABEL.
DCLCOPY	SP—TABLE/DCLCOPY.
DCLLBL	SP—TABLE/DCLLBL.
DCLRDEF	SP—TABLE/DCLRDEF.
COPY	SP—TABLE/COPY.
COPYLBL	SP—TABLE/COPYLBL.
COPYLV1	SP—TABLE/COPYLV1.
SYNONYM	SP—TABLE/SYNONYM.
JOINTBL	Identify Joined SQL tables. You can supply as many as 15 JOINTBLs. If you exceed this limit, the Key_Value in the Duplicate Key Error Message is replaced with the value 'LIMIT REACHED'.
JOINCOL	Identifies the columns used to join different SQL tables together. You can supply as many as 99,999 JOINCOLs for a JOIN. If you exceed this limit, the Key_Value in the Duplicate Key Error Message is replaced with the value 'LIMIT REACHED'.
TLNROW	Identify the Rows associated with a JOIN. You can supply as many as 99,999 TLNROWs for a JOIN. If you exceed this limit, the Key_Value in the Duplicate Key Error Message is replaced with the value 'LIMIT REACHED'.

JOIN/JOINTBL Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
JOINTBL	D	1	A	Y	1	
LABEL	P	8	A	Y	1	NAME
QUAL	P	8	A	Y	1	NAME
TBLNAME	P	18	A	Y	1	NAME

Transport identifier	Description
JOINTBL	Identifies another SQL table that will be joined together to form the SQL table being Transported. Do not supply a value on the JOINTBL Transport Record. If you supply a value, it is ignored.
LABEL	SP—TLNJOIN/FROM(CORRELATION NAME).
QUAL	SP—TLNJOIN/FROM(TABLE QUALIFIER).
TBLNAME	SP—TLNJOIN/FROM(TABLE NAME).

JOIN/JOINCOL Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
JOINCOL	D	1	A	Y	1	
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
JREF1LBL	P	8	A	Y	1	NAME
JREF1COL	P	18	A	Y	1	NAME
JREF2LBL	P	8	A	Y	1	NAME
JREF2COL	P	18	A	Y	1	NAME

Transport identifier	Description
JOINCOL	Do not supply a value on the JOINCOL Transport Record. If you supply a value, it is ignored.
LASTUSER	TDF—JOIN COLUMN/LAST USER TO UPDATE.
LASTUPD	TDF—JOIN COLUMN/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—JOIN COLUMN/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—JOIN COLUMN/DATE CREATED (YYMMDD).
JREF1LBL	SP—JOINCOL/REF1(TABLE LABEL).
JREF1COL	SP—JOINCOL/JREF1(COLUMN NAME).
JREF2LBL	SP—JOINCOL/JREF2(TABLE LABEL).
JREF2COL	SP—JOINCOL/JREF2(COLUMN NAME).

JOIN/TLNROW Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
TLNROW	D	8	A	Y	1	NAME
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
DCLCOL	D	18	A	N	99999	

Transport identifier	Description
TLNROW	SP-TLNROW/LABEL.
LASTUSER	TDF—SQL ROW/LAST USER TO UPDATE.
LASTUPD	TDF—SQL ROW/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—SQL ROW/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—SQL ROW/DATE CREATED (YYMMDD).
DCLCOL	SQL Column Definitions. You can supply as many as 99,999 DCLCOLs for a TLNROW. If you exceed this limit, the Key_Value in the Duplicate Key Error

Transport identifier	Description
	Message is replaced with the value 'LIMIT REACHED'.

JOIN/TLNROW/DCLCOL Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
DCLCOL	D	18	A	Y	1	NAME
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
ALIAS	P	30	A	N	1	NAME
KEY	P	2	A	N	1	
NOTNULL	P	1	A	N	1	YESNO
TYPE	P	8	A	Y	1	
LTH	P	4	N	Y	1	NUMERIC
DEC	P	2	N	N	1	NUMERIC
ACCESS	P	1	A	N	1	YESNO
JOINLBL	P	8	A	Y	1	NAME

Transport identifier	Description
DCLCOL	■ SP—DCLCOL/NAME ■ SP—DCLCOL/DCLHOST
LASTUSER	TDF—SQL COLUMN/LAST USER TO UPDATE.
LASTUPD	TDF—SQL COLUMN/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—SQL COLUMN/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—SQL COLUMN/DATE CREATED (YYMMDD).
ALIAS	SP—DCLCOL/ALIAS.
KEY	SP—DCLCOL/KEY.
NOTNULL	SP—DCLCOL/NOTNULL.

Transport identifier	Description
TYPE	SP—DCLCOL/TYPE. Values are: ■ 'INTEGER' ■ 'SMALLINT' ■ 'FLOAT' ■ 'DECIMAL' ■ 'CHAR' ■ 'VARCHAR' ■ 'LONGVARC' ■ 'GRAPHIC' ■ 'VARGRAPH' ■ 'LONGVARG' ■ 'DATE' ■ 'TIME' ■ 'TIMESTMP'
LTH	SP—DCLCOL/LTH.
DEC	SP—DCLCOL/DEC.
ACCESS	SP—DCLCOL/ACCESS.
JOINLBL	SP—DCLCOL/JOINLBL.

FILE Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
FILE	R	10	A	Y	1	
KEY	P	8	A	Y	1	NAME
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
DESC	P	40	A	N	1	

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
LABEL	P	8	A	N	1	NAME
COPY	P	8	A	N	1	NAME
COPYLV1	P	1	A	N	1	YESNO
COPYLBL	P	30	A	N	1	NAME
KEYLTH	P	3	N	N	1	NUMERIC
OPCODE	P	5	A	N	1	NAME
KEYNAME	P	60	A	N	1	
COBVSKY	P	60	A	N	1	OFNAME
TYPE	P	4	A	N	1	
ACCMODE	P	3	A	N	1	
INDEXOF	P	8	A	N	1	NAME
OPTLIST	P	60	A	N	1	
GENKEYL	P	30	A	N	1	
RECLTH	P	60	A	N	1	
LRECLMIN	P	5	N	N	1	NUMERIC
LRECLMAX	P	5	N	N	1	NUMERIC
BLKSIZE	P	5	N	N	1	NUMERIC
COBDIV	P	2	A	N	1	
OPEN	P	6	A	N	1	
REUSE	P	1	A	N	1	YESNO

Transport identifier	Description
FILE	Identifies the type of FILE that you are transporting. Values are: ■ 'SEQUENTIAL' ■ 'VSAM' Sequential and VSAM file names are mutually exclusive. If a FILE exists and is a different type than that being Transported_In, the Key_Value in the Duplicate Key Error Message contains the duplicate FILE Name.
KEY	SP—DATA SET/NAME.
LASTUSER	TDF—FILE/LAST USER TO UPDATE.

Transport identifier	Description
LASTUPD	TDF—FILE/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—FILE/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—FILE/DATE CREATED (YYMMDD).
DESC	TDF—DATA ADMINISTRATION MENU/DESC.
LABEL	SP—RECORD/LABEL.
COPY	SP—RECORD/COPY.
COPYLV1	SP—RECORD/COPYLV1.
COPYLBL	SP—RECORD/COPYLBL.
KEYLTH	SP—RECORD/KEYLEN.
OPCODE	SP—RECORD/OPCODE.
KEYNAME	SP—RECORD/KEY.
COBVSKY	SP—RECORD/COBVSKY.
TYPE	SP—DATA SET/ACCESS(VSAM Access Method) Identifies the VSAM Access Method. Values are: ■ 'KSDS' ■ 'RRDS' ■ 'ESDS'
ACCMODE	SP—DATA SET/ACCESS(Access Mode). Identifies the VSAM Access Method. Values are: ■ 'DYN' ■ 'RAN' ■ 'SEQ' ■ 'DIR'
INDEXOF	SP—DATA SET/INDEXOF.
OPTLIST	SP—RECORD/OPTLIST.
GENKEYL	SP—RECORD/GENKEYL.
RECLTH	TDF—UPDATE DATA SET DEFAULT DATA/RECLTH.
LRECLMIN	SP—DATA SET/LRECL(rec length or min rec length).
LRECLMAX	SP—DATA SET/LRECL(max rec length).
BLKSIZE	SP—DATA SET/BLKSIZE.
COBDIV	SP—RECORD/COBDIV.
OPEN	SP—DATA SET/OPEN. Values are:

Transport identifier	Description
	■ 'OUTPUT' ■ 'INPUT' ■ 'I O' ■ 'EXTEND' ■ 'UPDATE'
REUSE	■ TDF—UPDATE DATA SET DEFAULT DATA/REUSE ■ SP—DATA SET/REUSE

CICSQUE Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
CICSQUE	R	8	A	Y	1	
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
DESC	P	40	A	N	1	
LABEL	P	8	A	N	1	NAME
COPY	P	8	A	N	1	NAME
COPYLV1	P	1	A	N	1	YESNO
COPYLBL	P	30	A	N	1	NAME
TYPE	P	2	A	N	1	
LRECL	P	5	N	N	1	NUMERIC
SYSID	P	4	A	N	1	
MAIN	P	1	A	N	1	YESNO

Transport identifier	Description
CICSQUE	SP—CQUEUE/NAME.
LASTUSER	TDF—CICSQUE/LAST USER TO UPDATE.

Transport identifier	Description
LASTUPD	TDF—CICSQUE/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—CICSQUE/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—CICSQUE/DATE CREATED (YYMMDD).
DESC	TDF—DATA ADMINISTRATION MENU/DESC.
LABEL	SP—RECORD/LABEL.
COPY	SP—RECORD/COPY.
COPYLV1	SP—RECORD/COPYLV1.
COPYLBL	SP—RECORD/COPYLBL.
TYPE	SP—CQUEUE/TYPE.
LRECL	SP—CQUEUE/LRECL.
SYSID	SP—RECORD/SYSID.
MAIN	SP—CQUEUE/MAIN.

CICSJRNL Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
CICSJRNL	R	8	A	Y	1	
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
DESC	P	40	A	N	1	
LABEL	P	8	A	N	1	NAME
COPY	P	8	A	N	1	NAME
COPYLV1	P	1	A	N	1	YESNO
COPYLBL	P	30	A	N	1	NAME
LRECL	P	5	N	N	1	NUMERIC
JFILEID	P	2	N	N	1	NUMERIC
JTYPEID	P	2	A	N	1	

Transport identifier	Description
CICSJRNL	SP—CJOURNAL/NAME.
LASTUSER	TDF—CICSJRNL/LAST USER TO UPDATE.
LASTUPD	TDF—CICSJRNL/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—CICSJRNL/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—CICSJRNL/DATE CREATED (YYMMDD).
DESC	TDF—DATA ADMINISTRATION MENU/DESC.
LABEL	SP—RECORD/LABEL.
COPY	SP—RECORD/COPY.
COPYLV1	SP—RECORD/COPYLV1.
COPYLBL	SP—RECORD/COPYLBL.
LRECL	SP—CJOURNAL/LRECL.
JFILEID	SP—CJOURNAL/JFILEID.
JTYPEID	SP—CJOURNAL/JTYPEID.

PSB Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
PSB	R	8	A	Y	1	NAME
DESC	P	40	A	N	1	
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
LASTUSED	P	6	N	N	1	NUMERIC
LANG	P	5	A	N	1	
MAXQ	P	2	N	N	1	NUMERIC
CMPAT	P	3	A	N	1	
IOASIZE	P	5	N	N	1	NUMERIC

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SSASIZE	P	5	N	N	1	NUMERIC
IOEROPN	P	16	A	N	1	
OLIC	P	3	A	N	1	
PCB	D	8	A	N	99999	

Transport identifier	Description
PSB	■ SP—PSBGEN/PSBNAME ■ TDF—DATA ADMINISTRATION MENU/NAME
DESC	TDF—DATA ADMINISTRATION MENU/DESC.
LASTUSER	TDF—PSB/LAST USER TO UPDATE.
LASTUPD	TDF—PSB/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—PSB/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—PSB/DATE CREATED (YYMMDD).
LASTUSED	TDF—PSB/DATE LAST ACCESSED.
LANG	■ SP—PSBGEN/LANG ■ TDF—CREATE/UPDATE PSB, FILE GROUP/LANG Values are: ■ 'COBOL' ■ 'PLI' ■ 'ASSEM'
MAXQ	SP—PSBGEN/MAXQ.
CMPAT	■ SP—PSBGEN/CMPAT ■ TDF—CREATE/UPDATE PSB, FILE GROUP/COMPAT
IOASIZE	SP—PSBGEN/IOASIZE.
SSASIZE	SP—PSBGEN/SSASIZE.
IOEROPN	SP—PSBGEN/IOEROPN. The System performs no editing on the value supplied.
OLIC	SP—PSBGEN/OLIC.
PCB	Identifies the IMS databases to which the PSB is sensitive. For your book, remove this topic and replace it with your own topic objects.

Transport identifier	Description
	Ensure that all topics at level 1 in your TOC are based on a "chapter" or "appendix" template only. You can supply as many as 99,999 PCBs for a PSB. If you exceed this limit, the Key_Value in the Duplicate Key Error Message is replaced with the value 'LIMIT REACHED'.

PSB/PCB Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
PCB	D	8	A	Y	1	NAME
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
PCBTYPE	P	7	A	Y	1	
ALIAS	P	16	A	N	1	NAME
PROCOPT	P	8	A	N	1	
ALTRESP	P	3	A	N	1	
SAMETRM	P	3	A	N	1	
MODIFY	P	3	A	N	1	
EXPRESS	P	3	A	N	1	
LTERM	P	8	A	N	1	NAME
KEYLEN	P	3	N	N	1	NUMERIC
POS	P	1	A	N	1	
PROCSEQ	P	8	A	N	1	NAME
ABCALL	P	1	A	N	1	
PRINT	P	1	A	N	1	
SENSEG	D	8	A	N	99999	

Transport identifier	Description
PCB	■ SP—PCB/LTERM ■ SP—PCB/DBDNAME ■ SP—PCB/NAME ■ TDF—CREATE/UPDATE PSB/NAME
LASTUSER	TDF—PCB/LAST USER TO UPDATE.
LASTUPD	TDF—PCB/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—PCB/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—PCB/DATE CREATED (YYMMDD).
PCBTYPE	■ SP—PCB/TYPE ■ TDF—CREATE/UPDATE PSB, FILE GROUP/TYPE Values are: ■ 'DATABAS' ■ 'TPPCB' ■ 'GSAM'—
ALIAS	TDF—CREATE/UPDATE PSB, FILE GROUP/PCBNAME.
PROCOPT	■ SP—PCB/PROCOPT ■ TDF—CREATE/UPDATE PSB, FILE GROUP/PROCOPT The System performs no editing on the value supplied.
ALTRESP	SP—PCB/ALTRESP.
SAMETRM	SP—PCB/SAMETRM.
MODIFY	SP—PCB/MODIFY.
EXPRESS	■ SP—PCB/EXPRESS ■ TDF—CREATE/UPDATE PSB, FILE GROUP/EXPRESS
LTERM	TDF—CREATE/UPDATE PSB, FILE GROUP/LTERM.
KEYLEN	■ TDF—CREATE/UPDATE PSB, FILE GROUP/KEYLEN ■ SP—PCB/KEYLEN
POS	SP—PCB/POS. Values are: ■ 'M' ■ 'S'
PROCSEQ	SP—PCB/PROCSEQ.
ABCALL	TDF—CREATE/UPDATE PSB, FILE GROUP/ABCALL. Values are:

Transport identifier	Description
	■ 'Y' ■ 'N'
PRINT	TDF—CREATE/UPDATE PSB, FILE GROUP/PRINT. Values are: ■ 'Y' ■ 'N'
SENSEG	Identifies the IMS Segments to which the PSB is sensitive for an IMS DBD. For your book, remove this topic and replace it with your own topic objects. Ensure that all topics at level 1 in your TOC are based on a "chapter" or "appendix" template only. You can supply as many as 99,999 SENSEGs for a PCB. If you exceed this limit, the Key_Value in the Duplicate Key Error Message is replaced with the value 'LIMIT REACHED'.

PSB/PCB/SENSEG Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SENSEG	D	8	A	Y	1	NAME
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
PARENT	P	8	A	Y	1	NAME
PROCOPT	P	8	A	N	1	NAME
INDICES	P	58	A	N	1	
LABEL	P	8	A	N	1	

Transport Identifier	Description
SENSEG	■ SP—SENSEG/NAME ■ TDF—UPDATE SENSITIVE SEG/NAME
LASTUSER	TDF—SENSEG/LAST USER TO UPDATE.
LASTUPD	TDF—SENSEG/DATE LAST UPDATED (YYMMDD).

Transport Identifier	Description
UPDTIME	TDF—SENSEG/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—SENSEG/DATE CREATED (YYMMDD).
PARENT	SP—SENSEG/PARENT.
PROCOPT	■ SP—SENSEG/PROCOPT ■ TDF—UPDATE SENSITIVE SEG/PROCOPT. The System performs no editing on the value supplied.
INDICES	■ SP—SENSEG/INDICES ■ TDF—UPDATE SENSITIVE SEG/INDICES
LABEL	TDF—UPDATE SENSITIVE SEG/LABEL.

FILEGRP Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
FILEGRP	R	8	A	Y	1	NAME
DESC	P	40	A	N	1	
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
LASTUSED	P	6	N	N	1	NUMERIC
LANG	P	5	A	N	1	
CMPAT	P	3	A	N	1	
PCB	D	8	A	N	99999	
SQL	D	1	A	N	-	
DATA SET	D	8	A	N	-	
SENQUE	D	8	A	N	-	
SENJRNL	D	8	A	N	-	

Transport identifier	Description
FILEGRP	TDF—DATA ADMINISTRATION MENU/NAME.
DESC	TDF—DATA ADMINISTRATION MENU/DESC.
LASTUSER	TDF—File Group/LAST USER TO UPDATE.
LASTUPD	TDF—File Group/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—File Group/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—File Group/DATE CREATED (YYMMDD).
LASTUSED	TDF—File Group/DATE LAST ACCESSED.
LANG	TDF—CREATE/UPDATE PSB, FILE GROUP/LANG. Values are: ■ 'COBOL' ■ 'PLI' ■ 'ASSEM'
CMPAT	TDF—CREATE/UPDATE PSB, FILE GROUP/COMPAT.
PCB	Identifies the IMS Databases to which the File Group is sensitive. For your book, remove this topic and replace it with your own topic objects. Ensure that all topics at level 1 in your TOC are based on a "chapter" or "appendix" template only. You can supply a combined maximum of 99,999 PCBs, SQLs, and DATA SETs for a FILEGRP. If you exceed this limit, the Key_Value in the Duplicate Key Error Message is replaced with the value 'LIMIT REACHED'.
SQL	Identifies the DB2 Tables and Joined Tables to which the File Group is sensitive.
DATA SET	Identifies the Files to which the File Group is sensitive.
SENQUE	Identifies the CICS QUEUEs to which the File Group is sensitive.
SENJRNL	Identifies the CICS JOURNALs to which the File Group is sensitive.

FILEGRP/PCB Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
PCB	D	8	A	Y	1	NAME
LASTUSER	P	8	A	N	1	

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
PCBTYPE	P	7	A	Y	1	
ALIAS	P	16	A	N	1	NAME
PROCOPT	P	8	A	N	1	
ALTRESP	P	3	A	N	1	
SAMETRM	P	3	A	N	1	
MODIFY	P	3	A	N	1	
EXPRESS	P	3	A	N	1	
LTERM	P	8	A	N	1	NAME
KEYLEN	P	3	N	N	1	NUMERIC
POS	P	1	A	N	1	
PROCSEQ	P	8	A	N	1	NAME
ABCALL	P	1	A	N	1	
PRINT	P	1	A	N	1	
SENSEG	D	8	A	N	99999	

Transport identifier	Description
PCB	■ SP—PCB/LTERM ■ SP—PCB/DBDNAME ■ SP—PCB/NAME ■ TDF—CREATE/UPDATE PSB, FILE GROUP/NAME
LASTUSER	TDF—PCB/LAST USER TO UPDATE.
LASTUPD	TDF—PCB/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—PCB/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—PCB/DATE CREATED (YYMMDD).
PCBTYPE	■ SP—PCB/TYPE ■ TDF—CREATE/UPDATE PSB, FILE GROUP/TYPE

Transport identifier	Description
	Values are: ■ 'DATABAS' ■ 'TPPCB' ■ 'GSAM'
ALIAS	TDF—CREATE/UPDATE PSB, FILE GROUP/PCBNAME.
PROCOPT	■ SP—PCB/PROCOPT ■ TDF—CREATE/UPDATE PSB, FILE GROUP/PROCOPT The System performs no editing on the value supplied.
ALTRESP	SP—PCB/ALTRESP.
SAMETRM	SP—PCB/SAMETRM.
MODIFY	SP—PCB/MODIFY.
EXPRESS	■ SP—PCB/EXPRESS ■ TDF—CREATE/UPDATE PSB, FILE GROUP/EXPRESS
LTERM	TDF—CREATE/UPDATE PSB, FILE GROUP/LTERM.
KEYLEN	■ TDF—CREATE/UPDATE PSB, FILE GROUP/KEYLEN ■ SP—PCB/KEYLEN—
POS	SP—PCB/POS. Values are: ■ 'M' ■ 'S'
PROCSEQ	SP—PCB/PROCSEQ.
ABCALL	TDF—CREATE/UPDATE PSB, FILE GROUP/ABCALL. Values are: ■ 'Y' ■ 'N'
PRINT	TDF—CREATE/UPDATE PSB, FILE GROUP/PRINT. Values are: ■ 'Y' ■ 'N'
SENSEG	Identifies the IMS Segments to which the FILEGRP is sensitive for a PCB. You can supply as many as 99,999 SENSEGs for a PCB. If you exceed this limit, the Key_Value in the Duplicate Key Error Message is replaced with the value 'LIMIT REACHED'.

FILEGRP/PCB/SENSEG Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SENSEG	D	8	A	Y	1	NAME
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
PARENT	P	8	A	Y	1	NAME
PROCOPT	P	8	A	N	1	NAME
INDICES	P	58	A	N	1	
LABEL	P	8	A	N	1	

Transport identifier	Description
SENSEG	■ SP—SENSEG/NAME ■ TDF—UPDATE SENSITIVE SEG/NAME
LASTUSER	TDF—SENSEG/LAST USER TO UPDATE.
LASTUPD	TDF—SENSEG/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—SENSEG/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—SENSEG/DATE CREATED (YYMMDD).
PARENT	SP—SENSEG/PARENT.
PROCOPT	■ SP—SENSEG/PROCOPT ■ TDF—UPDATE SENSITIVE SEG/PROCOPT The System performs no editing on the value supplied.
INDICES	■ SP—SENSEG/INDICES ■ TDF—UPDATE SENSITIVE SEG/INDICES
LABEL	TDF—UPDATE SENSITIVE SEG/LABEL.

FILEGRP/SQL Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SQL	D	1	A	Y	1	
QUAL	P	8	A	Y	1	NAME
TBLNAME	P	18	A	Y	1	NAME
DB2TYPE	P	4	A	Y	1	
LABEL	P	8	A	Y	1	NAME
SENROW	D	8	A	N	99999	

Transport identifier	Description
SQL	Do not supply parameters on the SQL Transport Record. If you supply a value, it is ignored.
QUAL	SP—TABLE/QUAL.
TBLNAME	SP—TABLE/NAME.
DB2TYPE	Identifies the type of DB2/SQL Table. Values are: ■ BASE—Base DB2 table definition ■ JOIN—Joined DB2 table definition ■ VIEW—Joined DB2 table view
LABEL	SP—TABLE/LABEL
SENROW	Identifies the CA Telon Row(s) that you use from the DB2 Table identified by the QUAL and TBLNAME Transport Records. You can supply as many as 99,999 SENROWs for a SQL. If you exceed this limit, the Key_Value in the Duplicate Key Error Message is replaced with the value 'LIMIT REACHED'.

FILEGRP/SQL/SENROW Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SENROW	D	8	A	Y	1	NAME

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	6	N	N	1	NUMERIC
LABEL	P	8	A	Y	1	NAME
COLS	P	60	A	N	1	
COLMORE	P	1	A	N	1	YESNO
COLCOUNT	P	4	N	N	1	NUMERIC

Transport identifier	Description
SENROW	TDF—DELETE SENSITIVE TLNROWS/TLNROW.
LASTUSER	TDF—TLNROW/LAST USER TO UPDATE.
LASTUPD	TDF—TLNROW/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—TLNROW/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—TLNROW/DATE CREATED (YYMMDD).
LABEL	TDF—TLNROW/LABEL.
COLS	TDF—DELETE SENSITIVE TLNROWS/COLUMNS DEFINED.
COLMORE	TDF—DELETE SENSITIVE TLNROWS/MORE INDICATOR(+).
COLCOUNT	TDF—DELETE SENSITIVE TLNROWS/C-CNT.

FILEGRP/DATA SET Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
DATA SET	D	8	A	Y	1	NAME
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
CRTDATE	P	6	N	N	1	NUMERIC
ALIAS	P	16	A	N	1	NAME

Transport identifier	Description
DATA SET	TDF—CREATE/UPDATE PSB, FILE GROUP/NAME. Identifies the name of the Sequential or VSAM File that the File Group references.
LASTUSER	TDF—DATA SET/LAST USER TO UPDATE.
LASTUPD	TDF—DATA SET/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—DATA SET/TIME LAST UPDATED (HHMM).
CRTDATE	TDF—DATA SET/DATE CREATED (YYMMDD).
ALIAS	TDF—CREATE/UPDATE PSB, FILE GROUP/PCBNAME.

FILEGRP/SENQUE Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SENQUE	D	8	A	Y	1	NAME
LABEL	P	8	A	N	1	NAME

Transport identifier	Description
SENQUE	SP—CQUEUE/NAME
LABEL	

FILEGRP/SENJRNL Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
SENJRNL	D	8	A	Y	1	NAME
LABEL	P	8	A	N	1	NAME

Transport identifier	Description
SENJRNL	SP—CJOURNAL/NAME
LABEL	

PSTORE Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
PSTORE	R	8	A	Y	1	
DESC	P	40	A	N	1	
LASTUSER	P	8	A	N	1	
LASTUPD	P	6	N	N	1	NUMERIC
UPDTIME	P	4	A	N	1	
CRTDATE	P	8	N	N	1	NUMERIC
LASTUSED	P	8	N	N	1	NUMERIC
PSHV	D	1	A	N	-	

Transport identifier	Description
PSTORE	TDF—PROTOTYPING FACILITY MENU/PS NAME.
DESC	TDF—PRESENTATION STORE/DESCRIPTION.
LASTUSER	TDF—PRESENTATION STORE/LAST USER TO UPDATE.
LASTUPD	TDF—PRESENTATION STORE/DATE LAST UPDATED (YYMMDD).
UPDTIME	TDF—PRESENTATION STORE/TIME LAST UPDATED (HHMM).

Transport identifier	Description
CRTDATE	TDF—PRESENTATION STORE/DATE CREATED (YYMMDD).
LASTUSED	TDF—PRESENTATION STORE/LAST USER.
PSHV	Supplies the values contained in a Presentation Store. Presentation Store values are supplied in the same way that they are entered in the Presentation Store Editor.

PSTORE/PSHV Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
PSHV	D	1	A	Y	1	
DBNAME	P	30	A	N	1	
VALUE	P	30	A	N	20	
SUB	P	3	N	N	1	NUMERIC
LTH	P	3	N	N	1	NUMERIC

Transport identifier	Description
PSHV	Do not supply any values on this Record. If you supply values, they are ignored.
DBNAME	TDF—PRESENTATION STORE EDITOR/DBNAME.
VALUE	TDF—PRESENTATION STORE EDITOR/VALUE.
SUB	TDF—PRESENTATION STORE EDITOR/SUB.
LTH	TDF—PRESENTATION STORE EDITOR/LTH.

* (Comment) Object

Transport identifier	Record type	Field length	Data type	Required	Maximum # of occur.	FLDTYPE
*	D	71	A	N	1	

Transport identifier	Description
*	Supplies in-line comments in the Transport File. Comments must follow the TRANPORT Transport Record when they are used. You can supply a maximum of one Comment record before or after any object on the Transport File. Note: You can create comment objects for documentation purposes through user-written programs that create transport files. The data is not stored in any of the CA Telon databases.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

O

P

R

S

T

U

V

W

X