CA Telon® Application Generator

IDMS Database SQL Option Guide

r5.1

CA Product References

This document references the following CA products:

- CA Telon® Application Generator (CA Telon)
- CA IDMS™
- CA IDMS™ SQL
- CA Datacom®
- CA Datacom® 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 also available on the CA Support website, found at http://ca.com/docs.

Contents

Chapter 1: Introduction

This manual describes how to use the CA IDMS Database SQL Option with CA Telon.

To use this guide, you should be a skilled user of CA IDMS SQL and use this guide with the CA Telon system documentation.

You can find additional information about CA IDMS Database SQL Option in these manuals:

- *CA IDMS Database Administration Guide*
- *CA IDMS SQL Reference Guide*
- *CA IDMS/DB SQL Quick Reference Guide*
- *CA IDMS SQL Programming Guide*
- *CA IDMS SQL Self-Training Guide*

This chapter contains installation and customization instructions for the CA Telon CA IDMS SQL.

Installation Instructions

Install the CA Telon base product using the procedures described in the Installation Guide manual and include the password for CA IDMS. If the Base installation is complete, repeat the installation and select only CA IDMS.

Creating Job Streams and PROC JCL

This table lists INSTALL members used to create job streams and PROC JCL.

Member	Label	Activity
$JOB		Job stream JCL to execute PROC
$010		JCL procedure parameters
$020	SCRATCH1	Scratch step
$030	IEBUP1	Obtain CA Telon source and update MACLIB
$040	CONTROL	Create screen definition
$050	GEN	Assembly step; to generate COBOL or PL/I

Member	Label	Activity
		source code
$060	IEBUPSRC	Store temporary file
$070	PRINT	Print panel image
$080	RESOLVE	Resolve COPY statement
$090	DB2PC	DB2 precompiler step (DB2 only)
$100	ECP	ECP compiler step (CICS only)
$110	DCOMPRE	CA Datacom SQL precompiler step (CA Datacom only)
$120	ISQLDML	CA IDMS precompiler step[1]
$130	COB/PLI	COBOL compiler or PL/I editor step
$140	LKED	Linkage editor step
$150	BCF	CA IDMS SQL access module[1]
$160	DBRMSRC	Capture and print related source
$280	RESOLVE2	Resolve COPY statement[2]
$290	DB2PC2	DB2 precompiler step (DB2 CICS client only)
$300	IDMSDML2	CA IDMS precompiler step (CA IDMS CICS client only)
$310	ECP2	ECP compiler step[2]
$320	DCOMPRE2	CA Datacom SQL precompiler step (CA Datacom CICS client only)
$330	ISQLDML2	CA IDMS SQL precompiler step[3]
$340	COB2	COBOL compiler step[2]
$350	LKED2	Linkage editor step[2]
$360	BCF2	CA IDMS SQL access module step[3]
$370	DBRMSRC2	Capture and print related source[2]

[1] CA IDMS SQL only
[2] CICS client only
[3] CA IDMS SQL CICS only

#CUSTCTL Parameters

This table contains a brief description of the parameters used in #CUSTCTL for the CA IDMS SQL. You can find descriptions of the other parameters in the *Installation Guide*.

@ name	Change to
@COB2	Y
@ISSLIB	CA IDMS DBA LOADLIB
@ISBLIB	CA IDMS LOADLIB
@ISQLCTL	CA IDMS SYSCTL data set name
@ISCNTL	CA IDMS SYSIDMS data set name (can be @PDSQUAL.INSTALL if @PDSQUAL.INSTALL(SYSIDMS) is used)
@ISQLSYSCTL	CA IDMS SYSCTL ddname
@ISDMCL	CA IDMS default DMCL[1]
@ISDICTNM	CA IDMS default dictionary name[1]
@ISQL	'Y' to request the CA IDMS Database SQL Option

[1] in INSTALL member SYSIDMS

In CA Telon PWS, ensure that the environmental variable TNCOBOLC is set to IGYCRCTL.

Other INSTALL Members

The table below lists and describes the other members in INSTALL. These members are used only by CA IDMS SQL, except TRGDATA and TRGHELP. The chapters in this document containing instructions on how to use these members are also listed.

Member	Function	See
CTRISQLD	Load training data into a VSAM file and then into CA IDMS SQL tables, load help for training programs into a VSAM file	Model Training System (see page 23)
CTRISQPD	Create the physical database for the training tables	Model Training System (see page 23)
CTRISQSC	Create the schema for the training tables	Model Training System (see page 23)
CTRISQTB	Define the training tables to CA IDMS SQL	Model Training System (see page 23)
SQI200CL	JCL to precompile, compile, link, and create access module for the CA IDMS SQL extract	CA IDMS SQL Extract (see page 11)
SQLEXT	Execution JCL for the CA IDMS SQL extract	CA IDMS SQL Extract (see page 11)
SYSIDMS	Parameter file for the CA IDMS SQL precompiler and access module creation utility	#CUSTCTL Parameters (see page 9)
TRGDATA	Training data	CA IDMS SQL Extract (see page 11)
TRGHELP	Training help	CA IDMS SQL Extract (see page 11)

Chapter 2: CA IDMS SQL Extract

This chapter describes the CA IDMS SQL Extract Utility system used to extract table information from the mainframe and create a Transport file to be loaded into the TDF. This system also provides the ability to create COPY members from the tables selected.

The CA IDMS SQL Extract Utility is a batch process that allows you to extract SQL tables and views from a CA IDMS SQL catalog. The utility outputs the extracted table and column information in transport file format for processing by the Transport utility to populate the CA Telon Design Facility Data Administration.

For each extracted table, the utility also supports automatic creation of a copybook member containing COBOL descriptions of all columns in the table. You can name this copybook as a COPY member in CA Telon programs.

CA IDMS SQL Extract Utility processing is based on information specified on one or more control cards in an input file as described in Control Cards (see page 13). The utility produces a report providing a history of the control card processing.

System Components

The components for this system are:

- In the #PDSQUAL.INSTALL dataset, JCL members:
  - SQI200CL
  - SQLEXT
- A COBOL II source program, SQBPI200, in the #PDSQUAL.SOURCE dataset
- From the #PDSQUAL.LOAD dataset, load members:
  - SQBPI100
  - SQBPI300

System Usage

Job stream SQI200CL (in the JCL library) creates load module SQBPI200 by compiling COBOL program SQBPI200. SQI200CL:

- Precompiles the program using CA IDMS's precompiler IDMSDMLC
- Compiles and links the program using IBM COBOL II and the standard linkage editor
- Maintains the program's access module using CA IDMS's Batch Command Facility (IDMSBCF)

  Note: As any CA IDMS SQL program, an access module must be stored in the load area of each catalog against which the CA IDMS SQL Extract program executes.

The JCL member SQLEXT specifies the tables to extract. The user is required to supply information concerning COPY members and TDF data administration parameters such as TLNNAME.

Driver (SQBPM100)

This program specifies which tables are extracted from the CA IDMS SQL catalog.

The batch driver takes a series of input transactions specifying the selected tables, and optional information concerning copy members.

Once the program processes the input information, the program calls the SQBPI200 program to build the transport file and the copy members. The file, SQINTRAN, contains a series of 80-character input transactions that serve as input to the SQBPI100 program.

Build Transport (SQBPI200)

SQBPI100 calls this batch subroutine. The subroutine builds a transport file for all selected tables and produces a report summarizing the results. For more information about selecting tables for extraction, see Table Selection (see page 14).

For each table, if a COPY member is to be created, the subroutine makes a call to the SQBPM300 program.

Build COPY Members (SQBPI300)

SQBPI200 calls this batch subroutine to produce COPY members. The output sequential file for these members is produced containing IEBUPDT ./ ADD NAME= specifications that are used to load a PDS on the mainframe with individual members. For more information about requesting COPY members, see Copy Member (see page 16).

Control Cards

The control cards connect the batch job to the CA IDMS catalog, specify which tables to extract and if COPY or COPY members are created.

- CONNECT CONTROL (see page 13)—Connect to the CA IDMS catalog.
- Table Selection (see page 14)—Specify the CA IDMS SQL tables
- Copy Member (see page 16)— Specify the COPY members

This is a sample of the control cards required to extract a table and create a copybook member:

```
         1         2         3         4         5         6         7    8
123456789012345678901234567890123456789012345679012345678901234567890...0

USERID  CORPDBA       APPLDICT
TABLE   EMPDEMO       EMPLOYEE    DEMOEMP      DEMOEMPLDEMO    Y
TABLE   DEMO                  *                                         Y
TABLECPY EMPLCOPY DEMO-EMPL-TABLE        Y
```

CONNECT CONTROL

This control card identifies the user password. This is an optional control card used to identify the name of the user creating the extracted transport file and/or the dictionary name of the SQL Catalog containing the definitions to be extracted. If specified, it must be the first statement in the control card file.

Pos	Len	Description
1	8	CONTROL CARD TYPE—Specify the literal USERID to identify the connect control statement
9	18	USER ID[1]—Specify the User ID of user creating the extracted transport file. If omitted, defaults to CATIMPRT.
28	8	DICTNAME[1]—Specify the dictionary name of the CA IDMS SQL Dictionary containing the definitions. If omitted,defaults to the DICTNAME specified in SYSIDMS.

[1] Optional.

Connect Control Example

The following statement specifies CORPDBA is the user creating the extract and to extract tables from the APPLDICT dictionary:

```
         1         2         3         4         5         6         7    8
1234567890123456789012345678901234567890123456790123456789012345678901234567890...0

USERID CORPDBA                APPLDICT
```

Table Selection

This control card identifies the CA IDMS SQL tables or view you want to extract. You can specify up to 20 separate table selections.

Pos	Len	Description
1	8	CONTROL CARD TYPE—Specify the literal TABLE to identify a Table Selection statement.
9	18	SCHEMA—Specify the name of the SQL Schema for the table. This field cannot be blank.
28	18	TABLE—Specify the name of the SQL table or view to be extracted. This field cannot be blank. **Note:** To extract all tables defined in the schema named in the SCHEMA parameter, specify an "*" in position 28 of the TABLE parameter.
47	8	QUAL[1]—Specify the eight-byte CA Telon qualifier for the table. If omitted, defaults to the first eight bytes of the schema name specified in the SCHEMA parameter. **Note:** The combination of qualifier and table name must be unique for all tables in the TDF.
56	8	TLNNAME[1]—Specify the eight-byte CA Telon TLNNAME for the table. If omitted, defaults to the first eight bytes of the table name. The TLNNAME must be unique for all tables in the TDF.
65	4	PREFIX[1]—Specify a one- to four-byte prefix value concatenated with column name to provide an alias for each column of the selected table. If omitted, no alias is produced for the column name.

Pos	Len	Description
70	1	COPY[1]—Specify a "Y" in this field to produce a copybook member containing COBOL definitions for the table's columns.

[1] Optional

Table Selection Examples

The following statement requests extraction of the <DatacomIDMSExtract> table EMPDEMO.EMPLOYEE. The table definition is to be transported into the TDF with a qualifier of DEMOEMP and a TLNNAME of DEMOEMPL.

Each column in the table will have an alias consisting of the prefix DEMO concatenated to the column name; for example, DEMO.*column-name.*

```
         1         2         3         4         5         6         7    8
1234567890123456789012345678901234567890123456790123456789012345678901234567890...0

TABLE        EMPDEMO   EMPLOYEE   DEMOEMP   DEMOEMPL   DEMO   Y
```

This example requests extraction of all tables in the DEMO SQL schema and creation of a copybook member for each table.

- Each table is transported into the TDF with the default qualifier of DEMO
- The TLNNAME associated with each table defaults to the first eight bytes of the table's name
- The member name of the copybook defaults to the first eight bytes of the table's name

```
         1         2         3         4         5         6         7    8
1234567890123456789012345678901234567890123456790123456789012345678901234567890...0

TABLE   DEMO   Y
```

Copy Member

If you have specified a "Y" in the COPY parameter of a TABLE statement, you can supply this control card immediately following the TABLE statement to provide additional information about the copy member for the table.

Pos	Len	Description
1	8	CONTROL CARD TYPE—Specify the literal TABLECPY to identify a Copy Member statement.
10	8	MEMBER (Optional)—Specify the name of the copybook member for the generated ./ ADD NAME= statement. If omitted, defaults to the first eight bytes of the table name specified in the TABLE parameter of the Table Selection statement.
19	30	LABEL[1]—Specify the COBOL name of the group definition generated for the table. If omitted, no group level definition is produced in the copybook.
50	1	LEVEL 1[1]—Specify a "Y" in this field, if LABEL parameter has been specified and you want the group definition to contain a level of 01. If omitted, and LABEL has been specified, the group definition contains a level of 03.

[1] Optional

Copy Member Example

The following example produces an ./ADD NAME=EMPLCOPY statement for the copybook and generates a group statement "01 DEMO-EMPL-TABLE." before the elementary COBOL definitions for the columns in the selected table:

```
         1         2         3         4         5         6         7    8
12345678901234567890123456789012345678901234567901234567890123456789 0...0

TABLECPY   EMPLCOPY   DEMO-EMPL-TABLE               Y
```

Chapter 3: Loading the TDF with Programs and Tables

CA IDMS SQL tables created by the SQL Catalog Extract must be loaded into the TDF using the Transport utility. Also, there can be existing programs, in CA Telon source code format, that must be imported into the TDF. It is important to have agreement between the data administration specification in programs to be imported and the data administration definitions contained within the TDF.

Transport

The Transport files created by the SQL Catalog Extract are used to load the TDF data administration files with table information. Except for the ability to add TLNROW's exploiting existing column information in the tables, it is impossible to modify table information within the TDF.

For the CA Telon Training system, a Transport file is provided to pre-load the TDF prior to the importing of any Training system programs. This file is member CTRISQTP found within the SOURCE dataset.

Import

It is important to consider the data administration contained within the TDF before importing programs. Especially in the case of CA IDMS SQL, there can be differences in the area of TLNROW specification between what is contained in the TDF, and what might be in a particular program. Ensure the TDF data administration contains all TLNROW's that may apply to a particular table before importing programs specifying those TLNROW's. By use of the CTRISQTP member described above, the user can import all of the CA Telon Training system programs successfully.

Chapter 4: Program Generation

There are several issues to consider when generating programs for CA IDMS SQL. These issues include:

- Abend processing features
- Commit/rollback generation
- Access module control file
- Features

Abend Processing

When an unexpected return code occurs in a program containing CA IDMS SQL statements, in addition to the normal CA Telon abend processing, a PERFORM of a new section, Z-970-IDMSSQL-ERROR, is generated. Within this new section, a CALL is executed to the IDMSIN01 routine that loads the CA IDMS SQL error message into the SQLMSGS buffer.

For screen programs, this information is written to a CICS temporary storage queue enabling the user to view the message.

For batch programs, the contents of SQLMSGS are displayed to SYSOUT.

In both cases, the normal CA Telon abend processing is performed.

SQL Return Codes

The logic generated relative to SQL data access for CA IDMS SQL is handled in this manner:

- If the SQLCODE is found to be zero or +100, the value is moved into the <tablename>-STATUS field
- For other SQLCODE return codes (i.e., all negative values), the SQLCERC value is moved into <tablename>-STATUS

Hence, if you want to code explicit ignore codes, you must use the SQLCERC values for return codes other than zero (normal processing) or +100 (not found/end of file).

Note: These must be positive values.

The generic DA-STATUS code is loaded:

DA-STATUS	Codes
OK	0
NOTFOUND	+100
DUPLICATE	+1058 or +2212
LOGICERR	+6001
SECURITY	+1069 or +2365
NOTAVAIL	+2005 or +2006 or +2007
DBMERROR	(any other return code)

Commit/Rollback

For all programs containing CA IDMS SQL access, the U-100-COMMIT and U-100-ROLLBACK paragraphs are generated. For COMMIT, the syntax is:

```
EXEC SQL
COMMIT RELEASE
END-EXEC.
```

At the end of each screen program containing CA IDMS SQL access, there is an option to have CA Telon generate a PERFORM of U-100-COMMIT within the C-200-TERMIO-WRITE and the C-300-TERMIO-XFER sections.

If this perform is NOT desired, the IDMSCOM parameter must be set accordingly in the TLNIIS macro. The IDMSCOM parameter is located within the SETENV invocations for CICS or BATCH. Valid values for IDMSCOM are:

- Y—Default. Generates a perform of U-100-COMMIT
- N—Suppresses generation of the perform of U-100-COMMIT in sections:
  - C-200-TERMIO-WRITE
  - C-300-TERMIO-XFER

Access Module Control File

During the generation of CA IDMS SQL programs, a control file for access module processing is generated, and passed to the CA IDMS Batch Control Facility. The control records generated are:

```
DROP ACCESS MODULE <schema>.<program>;
COMMIT WORK;
CREATE ACCESS MODULE <schema>.<program> FROM <program>
    AUTO RECREATE ON;
```

Features

COBOL II is the only supported language for CA IDMS SQL.

CA Telon for CA IDMS SQL fully supports:

- 32-byte column names
- Creation and maintenance of CA Telon join tables
- ALIAS specification for tables and joins
- COPY specifications for table definitions

In generated programs when a COPY member is not specified for a table definition:

If no aliases are specified for the table's columns, CA Telon generates:

```
EXEC SQL INCLUDE TABLE <sql-schema>.<table-name>
AS <structure-name>
END-EXEC
```

Where:

<structure-name>

The host variable group name for the table developed by appending <table-name> to the value SQL. For example, if <table-name> is TRGEMPL, the <structure-name> is generated as SQLTRGEMPL.

If aliases are specified for the table's columns, CA Telon does not generate the AS <structure-name> clause.

This version supports the modification or display of column name ALIASes on the Add/Update SQL Alias screen, which you access from the Create/Update CA IDMS SQL Data Administration screen. In generated programs, the COPY specification for table definitions is ignored. CA Telon always generates:

```
EXEC SQL
INCLUDE TABLE <schema>.<table> AS <groupname>
END-EXEC
```

Where <groupname> is the host variable group name for the table.

Chapter 5: Model Training System

The model training system can be used for testing, training and demonstration of system operation accompanies the CA IDMS SQL installation. The CA IDMS SQL feature does not require the implementation of this training system, but the manner in which it is constructed and executed is educational. This section provides a description of the system components and instructions for operation.

System Overview

The model training system provides a series of screen and batch programs to operate upon a data structure related to employees within a company. These are the components of the data structure:

- Employee data—Information concerning the employee (name, address, telephone, etc.).
- Task/project data—Information concerning tasks and projects to which the employee is assigned.
- Time data—Information concerning time allotments by task/project for the employee.

This data is placed in several different files and tables for use by the supplied programs. The contents of each file and table are described below:

Object name	Object type	Contents
TREMPLB	INDEXED	Records for all three data components, with a flag for each record indicating which component is used
TRGEMPLC	CA IDMS SQL	Rows for employee data
TRGTASKC	CA IDMS SQL	Rows for task/project data
TRGTIMEC	CA IDMS SQL	Rows for time data
TRHELP	INDEXED	Help data for application

For file and table layouts of each of these objects, see File Layouts.

Prepare the CA IDMS Test Environment

Before generating CA Telon programs for CA IDMS SQL, define the training system's database and tables to CA IDMS. You must:

1. Define the segments that represent the physical database in CA IDMS's system dictionary; include the segments in the DMCL.
2. Create and format operating system files that will contain the training system's data.
3. Copy the segment definition from the system dictionary into the appropriate application dictionary.
4. Create the schema, tables and views in the CA IDMS application catalog.

These members, in the INSTALL dataset, will assist you in accomplishing these tasks:

Member	Function
CTRISQPD	Contains the physical database definition and DMCL for the training tables
CTRISQSC	Contains the schema and physical database definitions for the application catalog
CTRISQTB	Contains the DDL statements defining the SQL tables and indexes

For detailed discussions on CA IDMS's physical database and CA IDMS SQL schema and table definitions, refer to:

- *CA IDMS Database Administration Guide*
- *CA IDMS Database Design Guide*
- *CA IDMS Database SQL Option Guide*

Physical Database Definitions

In the supplied training system, the CTRISQPD member in the INSTALL dataset contains suggested CA IDMS physical database (SEGMENT, FILE, AREA, and DMCL) definitions and alterations.

The CTRISQPD member contains a default DMCL name of CVDMCL. If this is not the name of your DMCL, edit CTRISQPD changing all occurrences of CVDMCL to your DMCL name.

The CTRISQPD member is used as input to the CA IDMS batch command facility (IDMSBCF). This job:

1. Adds the TELON segment definition to the catalog containing the DMCL definition.
2. Modifies the DMCL to include the TELON segment.
3. Generates and punches the DMCL for link editing.

Note: The CTRISQPD member does NOT create a separate BUFFER definition for the TELON segment. If the DMCL does not specify a 'DEFAULT BUFFER', either:

- Edit the CTRISQPD and add a BUFFER definition **or**
- Modify your DMCL to specify a default buffer

Create and Format Files for the CA IDMS Database

To define the CA Telon model training database in CA IDMS, you must allocate a dataset for the training system CA IDMS database (100 4276-byte pages). Using the DMCL modified in Physical Database Definitions (see page 24) and the CA IDMS Command Facility to initialize the database with the FORMAT statement:

```
FORMAT FILE TELON.TRGIDSQL;
```

Create the Schema and Database Definitions

The CTRISQSC member in the INSTALL dataset contains suggested CA IDMS definitions for the:

- Schema
- Physical database
  - SEGMENT
  - FILE
  - AREA

These statements are used to update the application dictionary with the needed database information.

SQL Table Definitions

The CTRISQTB member in the INSTALL dataset contains the definitions for:

- Table definitions
- Indexes
- Views

Programs

The programs supplied perform a variety of functions relative to these objects in both online and batch modes of operation.Therefore, CA Telon source with the name TRCCSASD produces a COBOL program named TRCPCCSA, while CA Telon source with the name TRS500BD produces a COBOL program named TRBPS500.

Screen Programs

The screen programs are supplied with ID's of the form CCM*i*. The *i* code within the ID represents these programs:

***i* Code**	**Data**	**Program Description**
A	Employee	Add/update screen. Allows the user to add new employees, or update existing employee information.
C	Employee	Combination screen—add, update, delete, inquiry. Allows the user to preform all maintenance functions for the employee data within a single program.
D	Employee	Display screen. Displays employee data.
E	None	Exit screen. Application termination screen.
H	HELP	HELP screen. Provides field and screen level HELP.
L	Employee	List screen. Lists employee data with row incrementation.
M	Employee	Menu screen. Application starting point.
N	Employee	Non-terminal. CICS non-terminal program providing report of employee data.
P	Task/ project	Task/project screen. Maintenance for the task/project data.
T	Time	Time screen. Maintenance for the time data.

i Code	Data	Program Description
X	None.	Screen program which initiates the **N** non-terminal program.
Z	Employee	Zap screen. Deletes employee records.

For layout of TRHELP, see File Layouts.

Note: The Help program is TRCCVH, which is used by all training systems.

Batch Programs

There are three batch programs of the form M*i*00.

i Code	Program Description	Input Data	Output Data
S100	Create SQL data. After deleting contents, takes indexed TRGEMPLB file and reloads CA IDMS SQL tables[1]: ■ TRGEMPL ■ TRGTASK ■ TRGTIME	TRGEMPLB	TRGEMPL TRGTASK TRGTIME
S500	Create TRGEMPLB data. Takes sequential data contained in file TRGDATA and reloads the TRGEMPLB indexed file[1]	TRGDATA	TRGEMPLB
S700	Create TRHELP data. Takes sequential data contained in file TRGHELP and reloads the TRHELP indexed file[1]	TRGHELP	TRHELP

[1] Also provides a report of the processing

To refresh the model training databases, including reloading the HELP and CA IDMS SQL data, submit these job streams in order:

- S500
- S100
- S700

Building the Model Training System

To use the model training system, the application programs must be compiled and linked, the training data must be loaded into the database, and the CICS tables must be updated.

Program Generation and Compilation

The SOURCECC dataset provides the source code for the model training system. These program definitions can be imported into the TDF to browse, modify, and generate the actual COBOL II programs.

Online Programs—CA IDMS SQL Access

The INSTALL dataset provides generate, compile, and link JCL procedures and job streams. Job streams JSCPG2L and JSCXG2L, which invoke procedures TLSCPG2L and TLSCXG2L, are used to generate, compile, and link the training system screen programs. These programs contain CA IDMS SQL access from either CA Telon source (JSCPG2L), or can be accessed directly from the TDF (JSCXG2L)) after the programs have been imported.

Batch Programs—CA IDMS SQL Access

Similarly, job streams JSBPG2L and JSBXG2L, which invoke procedures TLSBPG2L and TLSBXG2L, are used to generate, compile, and link the training systems batch programs. These programs contain CA IDMS SQL statements from either CA Telon source (JSBPG2L), or can be accessed directly from the TDF (JSBXG2L) after the programs have been imported.

Programs—No CA IDMS SQL Access

Job streams JNBPG2L, JNBXG2L, JNCPG2L, and JNCXG2L and their respective procedures, provide similar functionality as Online Programs—CA IDMS SQL Access and Batch Programs—CA IDMS SQL Access, except without the CA IDMS SQL access processing.

Summary of Programs

This is a summary of the jobs streams and procedures used with the training system:

Program ID	Job streams	Procedures	Comments
CCSA	JSCPG2L,	TLSCPG2L,	Add/update screen

Program ID	Job streams	Procedures	Comments
	JSCXG2L	TLSCXG2L	
CCSC	JSCPG2L, JSCXG2L	TLSCPG2L, TLSCXG2L	Combination screen
CCSD	JSCPG2L, JSCXG2L	TLSCPG2L, TLSCXG2L	Display screen
CCSE	JNCPG2L, JNCXG2L	TLSCPG2L, TLSCXG2L	Exit screen
CCVH	JNCPG2L, JNCXG2L	TLSCPG2L, TLSCXG2L	HELP screen
CCSL	JSCPG2L, JSCXG2L	TLSCPG2L, TLSCXG2L	List screen
CCSM	JSCPG2L, JSCXG2L	TLSCPG2L, TLSCXG2L	Menu screen
CCSN	JSCPG2L, JSCXG2L	TLSCPG2L, TLSCXG2L	Non-terminal program. Requires specifying parameters: ■ DEFTYPE=ND ■ OPTION=3 ■ ECPPARM='SP,SOURCE'
CCSP	JSCPG2L, JSCXG2L	TLSCPG2L, TLSCXG2L	Task/project screen
CCST	JSCPG2L, JSCXG2L	TLSCPG2L, TLSCXG2L	Time screen
CCSX	JNCPG2L, JNCXG2L	TLSCPG2L, TLSCXG2L	Screen program which initiates the **CCSN** non-terminal program
CCSZ	JSCPG2L, JSCXG2L	TLSCPG2L, TLSCXG2L	Zap screen
S100	JSBPG2L, JSBXG2L	TLSBPG2L, TLSBXG2L	Create SQL data batch program
S500	JSBPG2L, JSBXG2L	TLSBPG2L, TLSBXG2L	Create TRGEMPLB data batch program
S700	JSBPG2L, JSBXG2L	TLSBPG2L, TLSBXG2L	Create HELP data batch program

Note: Various COPY members are contained within the SOURCECC dataset, requiring CPYLIB be assigned to SOURCECC. Copies of data for loading training data are found in members TRGDATA and TRGHELP within the INSTALL dataset; for more information, see Loading Data (see page 30).

Assemble BMS Maps

The INSTALL member JCPBMS, which calls PROC TLNCPBMS, is used to assemble the BMS maps for the screen programs listed in Summary of Programs (see page 28). It must be run for these program IDs:

CCSA
CCSC
CCSD
CCSE
CCSH
CCSL
CCSM
CCSP
CCST
CCSX
CCSZ

Re-assemble CICS Tables

The PCT and PPT entries for the training system are found in the TABLES dataset under the names TRPCTCS and TRPPTCS. These copy members must be added to existing table source; afterwards, the table source must be re-assembled.

Note: Define the TRHELP dataset to CICS as a VSAM file.

Loading Data

After the model training system programs TRS100BD, TRS000BD, TRS500BD, and TRS700BD have been generated, compiled and linked, and the CA IDMS SQL tables defined, follow these steps to populate the tables with data:

1. The source for the training data is found in #PDSQUAL.INSTALL members TRGDATA and TRGHELP. Both members must be copied into sequential datasets with their respective characteristics before the data is loaded.

Member	Name	Characteristics	
TRGDATA	#PDSQUAL.TRGDATA	organization record length record format blocksize	= PS = 60 = FB = n[1] X 60

Member	Name	Characteristics	
TRGHELPM	#PDSQUAL.TRGHELP	organization record length record format blocksize	= PS = 44 = FB = n[1] X 44

[1] n is the appropriate blocking factor for your site.

2. The actual load is done by #PDSQUAL.INSTALL (CTRISQLD) is used to load the SQL tables with data (and also the HELP file).

Executing the CICS System

The execution of the model training system is accomplished by the specification of the CICS transaction code for the top level menu program, TCSM.

Note: Each screen program has its own transaction code of the form TCS*i*, where *i* = A, C, D, E, H, L, M, P, T, X, or Z.

File Layouts

This section provides details about the following file layouts:

- TRGEMPLB
  - TRGEMPL Record Layout
  - TRGTASK Record Layout
  - TRGTIME Record Layout
- TRGEMPL
- TRGTASK
- TRGTIME
- TRHELP

TRGEMPLB

```
         04  EMPL-KEY-AREA.
             05  EMPL-KEY.
                 10  EMPL-ID           PIC X(6).
             05  TASK-KEY.
                 10 TASK-PROJ-ID       PIC X(4).
                 10 TASK-TASK-ID       PIC X(4).
             05  TIME-KEY.
                 10 TIME-YEAR          PIC 9(2).
         04  EMPLOYEE-AREA     PIC X(583).
       04  RECORD-TYPE-INDICATOR   PIC X.

************************************************************************ *
*    TYPE OF RECORD IS DETERMINED BY THE VALUE OF THE                    *
*    RECORD-TYPE-INDICATOR:     1 = TRGEMPL RECORD                       *
*                                       2 = TRGTASK RECORD               *
*                                       3 = TRGTIME RECORD               *
************************************************************************ *
************************ LENGTH OF EACH RECORD IS 600 BYTES ************* *
```

TRGEMPL Record Layout

```
************************************************************************ *
*    TRGEMPL RECORD LAYOUT                                               *
************************************************************************ *
         04 TRGEMPL-AREA REDEFINES EMPLOYEE-AREA.
             05 EMPL-NAME              PIC X(25).
             05 EMPL-DOB               PIC 9(6).
             05 EMPL-SEX               PIC X.
             05 EMPL-PHONE             PIC X(10).
             05 EMPL-STREET            PIC X(25).
             05 EMPL-CITY              PIC X(25).
             05 EMPL-STATE             PIC XX.
             05 EMPL-ZIP               PIC X(5).
             05 EMPL-DOE               PIC 9(6).
             05 EMPL-DEPARTMENT PIC XXX.
             05 EMPL-HOURLY-RATE       PIC 999V99 COMP-3.
             05 EMPL-HOURS             PIC 999V9.
             05 FILLER         PIC X(468).
************************************************************************ *
*    END OF TRGEMPL RECORD LAYOUT                                        *
************************************************************************ *
```

TRGTASK Record Layout

```
************************************************************************ *
*    TRGTASK RECORD LAYOUT                                                *
************************************************************************ *
         04 TRGTASK-AREA REDEFINES EMPLOYEE-AREA.
             05 TASK-PROJ-STUFF.
                 10 TASK-P-DESC         PIC X(22).
                 10 TASK-P-PRTY PIC X(2).
                 10 TASK-P-CODE PIC X(6).
             05 TASK-TASK-STUFF.
                 10 TASK-T-DESC PIC X(22).
                 10 TASK-T-PRTY PIC X(2).
                 10 TASK-T-CODE PIC X(6).
             05 FILLER          PIC X(523).
************************************************************************ *
*    END OF TRGTASK RECORD LAYOUT                                         *
************************************************************************ *
```

TRGTIME Record Layout

```
************************************************************************ *
*    TRGTIME RECORD LAYOUT                                                *
************************************************************************ *
         04 TRGTIME-AREA REDEFINES EMPLOYEE-AREA.
             05 TIME-FOR-BATCH.
                10 TIME-BY-QUARTER      OCCURS 4  TIMES.
                   15 TIME-WEEKLY       OCCURS 13 TIMES.
                      20 TIME-REG       PIC S9(3)V9 COMP-3.
                      20 TIME-OT        PIC S9(3)V9 COMP-3.
                10 TIME-TOTALS  OCCURS 4 TIMES.
                   15 TIME-REG-QUARTERLY-TOTAL
       PIC S9(5)V9 COMP-3.
             05 FILLER          PIC X(255).
************************************************************************ *
*    END OF TRGTIME RECORD LAYOUT                                         *
************************************************************************ *
```

TRGEMPL

```
01  SQLTRGEMPLC.
    01  SQLTRGEMPLC.
    05  EMPL-ID                 PIC X(6).
    05  EMPL-NAME               PIC X(25).
    05  EMPL-DOB                        PIC S9(6).
    05  EMPL-SEX                        PIC X(1).
    05  EMPL-PHONE              PIC X(10).
    05  EMPL-STREET             PIC X(25).
    05  EMPL-CITY                       PIC X(25).
    05  EMPL-STATE              PIC X(2).
    05  EMPL-ZIP                        PIC X(5).
    05  EMPL-DOE                        PIC S9(6).
    05  EMPL-DEPARTMENT         PIC X(3).
    05  EMPL-HOURLY-RATE                PIC S9(3)V9(2).
    05  EMPL-HOURS              PIC S9(3)V9(1).
```

TRGTASK

```
01  SQLTRGTASKC.
    05  TASK-EMPLID            PIC X(6).
    05  TASK-PROJID            PIC X(4).
    05  TASK-TASKID            PIC X(4).
    05  TASK-P-DESC            PIC X(22).
    05  TASK-P-PRTY            PIC X(2).
    05  TASK-P-CODE            PIC X(6).
    05  TASK-T-DESC            PIC X(22).
    05  TASK-T-PRTY            PIC X(2).
    05  TASK-T-CODE            PIC X(6).
```

TRGTIME

```
01  SQLTRGTIMEC.
    05  TIME-EMPL-ID           PIC X(6).
    05  TIME-PROJ-ID           PIC X(4).
    05  TIME-TASK-ID           PIC X(4).
    05  TIME-YEAR                      PIC S9(2).
    05  TIME-C-1-26            PIC X(156).
    05  TIME-C-27-52           PIC X(156).
    05  TIME-Q-TOT1            PIC S9(5)V9(1).
    05  TIME-Q-TOT2            PIC S9(5)V9(1).
    05  TIME-Q-TOT3            PIC S9(5)V9(1).
    05  TIME-Q-TOT4            PIC S9(5)V9(1).
```

TRHELP

```
03  TRHELP-AREA                PIC X(1058).
 03  FILLER REDEFINES TRHELP-AREA.
       05  HELP-KEY            PIC X(8).
       05  HELP-MSG.
           10  HELP-MSG-LINE   OCCURS 15 TIMES
                INDEXED BY HDX PIC X(70).
```

Index