

**CA IDMS™**

## **ASF User Guide**

**Release 18.5.00, 2nd Edition**



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

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

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

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

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

The manufacturer of this Documentation is CA.

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

Copyright © 2014 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

## CA Technologies Product References

This guide references the following CA products

- CA OLQ™ Online Query for CA IDMS™
- CA Culprit™ for CA IDMS™
- CA IDMS™/DC
- CA IDMS™ UCF
- CA ADS™

## Contact CA Technologies

### Contact CA Support

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

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

### Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to [techpubs@ca.com](mailto:techpubs@ca.com).

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

## Documentation Changes

The following documentation updates were made for the 18.5.00, 2nd Edition release of this documentation:

- [Input Parameter Statements](#) (see page 273)—Added several statements.

# Contents

---

<b>Chapter 1: Introduction</b>	<b>13</b>
Syntax Diagram Conventions .....	13
<b>Chapter 2: The Automatic System Facility</b>	<b>17</b>
Data Tables .....	19
Stored Tables .....	21
Non-SQL Defined Records .....	22
Views .....	22
Relational Operations.....	23
ASF in the Corporate Environment .....	25
ASF in the Information Center .....	25
ASF in the Development Center .....	27
<b>Chapter 3: Using ASF</b>	<b>29</b>
The Structure and Functions of ASF.....	30
Session Control.....	32
Information .....	33
Query.....	33
Definition.....	33
Data Access .....	34
Passkey.....	35
Administrative.....	36
ASF Screen Format .....	37
ASF Screen Management Features .....	38
Function Key Assignments Are Predefined .....	38
Cursor Movement .....	39
Message Display and Field Highlighting.....	39
Default Value Assignment .....	40
Conducting an ASF Session.....	40
Beginning an ASF session.....	40
Moving Between ASF Screens.....	41
Ending an ASF Session.....	43
Selecting Tables.....	43

---

## Chapter 4: Creating and Using Stored Tables 47

Outlining an Application .....	48
Step 1: Determining the Required Tables .....	48
Step 2: Defining the Columns for Each Table .....	48
Step 3: Specifying the Unique Keys .....	50
Step 4: Establishing Join Columns .....	50
Defining a Stored Table.....	51
Step 1: Name the Table.....	51
Step 2: Defining the Columns.....	53
Step 3: Generating the Table .....	57
Using a Data Table .....	59
Step 1: Loading Data .....	59
Step 2: Displaying Data.....	62
Step 3: Changing Data .....	62
Step 4: Deleting Data .....	63
Step 5: Selecting Data.....	63
Step 6: Displaying a Table Through CA O LQ .....	66
Stored Table Summary .....	69

## Chapter 5: Deriving Views 73

Defining Views .....	74
Step 1: Naming the View .....	74
Step 2: Specifying the Source Tables and Join Criteria .....	76
Step 3: Choosing the Columns .....	77
Step 4: Adding a Column.....	81
Step 5: Selecting Rows .....	83
Step 6: Generating the View .....	85
Using Views .....	86
Step 1: Displaying Rows .....	86
Step 2: Selecting Rows .....	87
Step 3: Changing Rows .....	89
View Summary .....	90

## Chapter 6: Granting Access to Tables 91

Assigning Passkeys.....	92
Giving Passkeys.....	92
Displaying Passkeys.....	97
Deleting Passkeys.....	99
Passkey Summary.....	100

---

<b>Chapter 7: Restricting Access to Rows</b>	<b>101</b>
Assigning Row-level Security to Data Tables.....	102
Assigning Row-level Security.....	103
Row-level Security Summary .....	112
<b>Chapter 8: ASF Processing</b>	<b>115</b>
Table Definition.....	115
Table Storage.....	118
Table Generation.....	119
<b>Chapter 9: Extended Functions of ASF</b>	<b>123</b>
Specifying Underlying Table Characteristics .....	123
Assigning a Table to an Area .....	124
Storing a Table in an Area .....	125
Limiting the Use of a Table.....	126
Defining Column Attributes.....	127
Specifying Column Formats.....	128
Defining Column Levels and Values.....	130
Examples.....	133
Mapping Considerations.....	135
Selecting Data at Definition Time.....	137
Deriving Views from Various Sources.....	139
Deriving Views from Data Tables .....	141
Deriving Views from Non-SQL Defined Records .....	142
Examples.....	145
Creating Indexes.....	150
Fast Access Keys .....	152
Additional Index Keys .....	152
Defining the Storage and Display Sequence.....	153
<b>Chapter 10: Redefining, Regenerating, and Deleting Tables</b>	<b>155</b>
Redefining Tables.....	155
Regenerating Tables .....	158
Unsuccessful Regenerate.....	159
Deleting Tables .....	160
Modifying and Deleting Source Tables.....	161
<b>Chapter 11: Administrative Functions of ASF</b>	<b>165</b>
Initializing the Definition Area .....	166

---

The Steps of Initialization.....	166
Specifying the Default Area for the System.....	167
Acquiring and Distributing Administrative Authority .....	168
Specifying System and User Defaults .....	169
Defaults for Models Used by ASF .....	169
Performing General ASF Administration.....	171
Adding Secondary Areas .....	171
Unlocking Table Definitions.....	172
Restricting ASF Users .....	172
Limiting Passkey Authorization.....	172
Limiting Row-level Security Authorization.....	173
Locking and Unlocking Users.....	178
Deleting Users from the Catalog .....	178
Deleting All Object Entities in the Catalog.....	179

## **Chapter 12: Accessing Tables Outside ASF** **181**

Information Center Management System .....	182
CA OLQ .....	182
CA Culprit.....	183
CA IDMS/DB Data Manipulation Language (DML) .....	183
Application Development System (CA ADS) .....	184

## **Chapter 13: Basic Screens** **185**

Signon Screen—SIGN.....	186
Activity Selection Screen—ASEL .....	188
Table Definition Screen—TDEF .....	191
Column Definition Screen—CDEF .....	196
Data Selection Screen—DSEL .....	200
Examples.....	206
Data Display Screen—DLOD (or) DDIS .....	209
Message Screen—MSGS .....	211
Help Screens—HELP .....	212

## **Chapter 14: Passkey Screens** **215**

Passkey Security .....	216
Data Access Passkeys.....	217
Catalog Access Passkeys .....	218
Predefined Passkey Combinations .....	219
Passkey.....	220
Give Passkeys Screen—GPAS .....	222



---

Delete Passkeys Screen—DPAS .....	223
Show Passkeys Given Screen—PASG .....	225
Passkeys Given Screen—(No screen name) .....	226
Show Passkeys Held Screen—PASH .....	227
Passkeys Held Screen—(no screen name) .....	228

## **Chapter 15: Row-Level Security Screens** **231**

Row-level Security.....	231
System Table Maintenance Screen—SYST .....	232
Row-level Security Screen—RSEC .....	233
Load Row-level Security Screen—RLOD .....	235
Row-level Security Selection Screen—RSEL .....	236
Display Row-level Security Screen—RDIS .....	238

## **Chapter 16: Extended Screens** **241**

Extended Table Definition Screen—XTAB .....	241
Extended Column Definition Screen—XCOL .....	245
Extended Selection Definition Screen—XSEL .....	250
Extended Key Definition Screen—XKEY .....	251
Extended Table Derivation Screen—XDER .....	254

## **Chapter 17: Administrative Screens** **259**

Administrator Control Screen—CNTL .....	259
Catalog Control Screen—CTLG .....	262
User Limits and Defaults Screen—ULIM .....	263
User Maintenance Screen—USER .....	267
System Table Maintenance Screen—SYST .....	268

## **Chapter 18: IDMSRADM** **271**

Description .....	271
Syntax .....	271
Input Parameter Statements .....	273
SIGNON Statement .....	274
SET AREA Statement .....	275
SET OWNER Statement .....	275
SET COMMIT Statement .....	276
IMPORT Statement .....	277
EXPORT Statement .....	279
DELETE Statement .....	281

---

PROCESS Statement.....	283
REPORT Statement.....	284
ARCHIVE Statement.....	285
SIGNOFF/BYE Statement.....	285
General Usage.....	286
JCL.....	287
z/OS—Central Version (CV).....	288
z/OS—Local Mode.....	289
z/VSE—Local Mode.....	290
z/VSE—Central Version (CV).....	292
z/VM—Local Mode.....	293
z/VM—Central Version (CV).....	294
IDMSLBS Procedure.....	295

## **Chapter 19: MAINTAIN ASF** **301**

Description.....	301
Usage.....	305
The Steps of MAINTAIN ASF.....	306
Sample Output.....	309
JCL.....	310
z/OS.....	311
z/VSE—Local Mode.....	313
z/VM—Local Mode.....	315
z/VM—Central version (CV).....	316
Examples.....	317

## **Appendix A: Logical Record Facility** **319**

Introduction.....	319
Checking Path Statuses.....	321
Examples.....	322
Example 1—Allowing for CALC Database Entry.....	322
Example 2—Adding Path-groups to Expand Functionality.....	325
Example 3—Implementing Referential Integrity Checks for a Table.....	329
Example 4—Performing a Difference Join.....	331
Example 5—Calculating Summary Information.....	335

## **Appendix B: Performing UNLOAD/RELOAD Against an ASF Database** **341**

Procedure.....	341
Preparing Your Environment.....	342
Running the IDMSRSSM Program.....	342

---

IDMSRSSM Execution JCL.....	343
Output from IDMSRSSM .....	346
UNLOAD and RELOAD of ASF Data Areas.....	349
UNLOAD and RELOAD ASF Definition Areas.....	349
Running the IDMSRUPD Program.....	349
IDMSRUPD Execution JCL.....	350
Output from IDMSRUPD .....	353

**Index**

**355**



# Chapter 1: Introduction

---

This manual applies to the CA IDMS Database Automatic System Facility (ASF). It serves as a user guide to creating data tables and developing applications through the CA IDMS Database Automatic System Facility.

This guide is intended for two groups of ASF users:

- End users who have little or no data processing experience
- Experienced data processing professionals, including the database administration staff

## Syntax Diagram Conventions

The syntax diagrams presented in this guide use the following notation conventions:

**UPPERCASE OR SPECIAL CHARACTERS**

Represents a required keyword, partial keyword, character, or symbol that must be entered completely as shown.

**lowercase**

Represents an optional keyword or partial keyword that, if used, must be entered completely as shown.

*italicized lowercase*

Represents a value that you supply.

**lowercase bold**

Represents a portion of the syntax shown in greater detail at the end of the syntax or elsewhere in the document.

←

Points to the default in a list of choices.

▶—————

Indicates the beginning of a complete piece of syntax.

—————▶

Indicates the end of a complete piece of syntax.

—————→

Indicates that the syntax continues on the next line.

▶—————

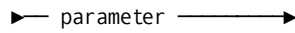
Indicates that the syntax continues on this line.



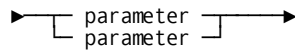
Indicates that the parameter continues on the next line.



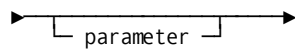
Indicates that a parameter continues on this line.



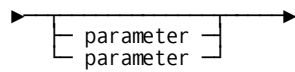
Indicates a required parameter.



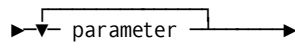
Indicates a choice of required parameters. You must select one.



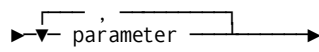
Indicates an optional parameter.



Indicates a choice of optional parameters. Select one or none.



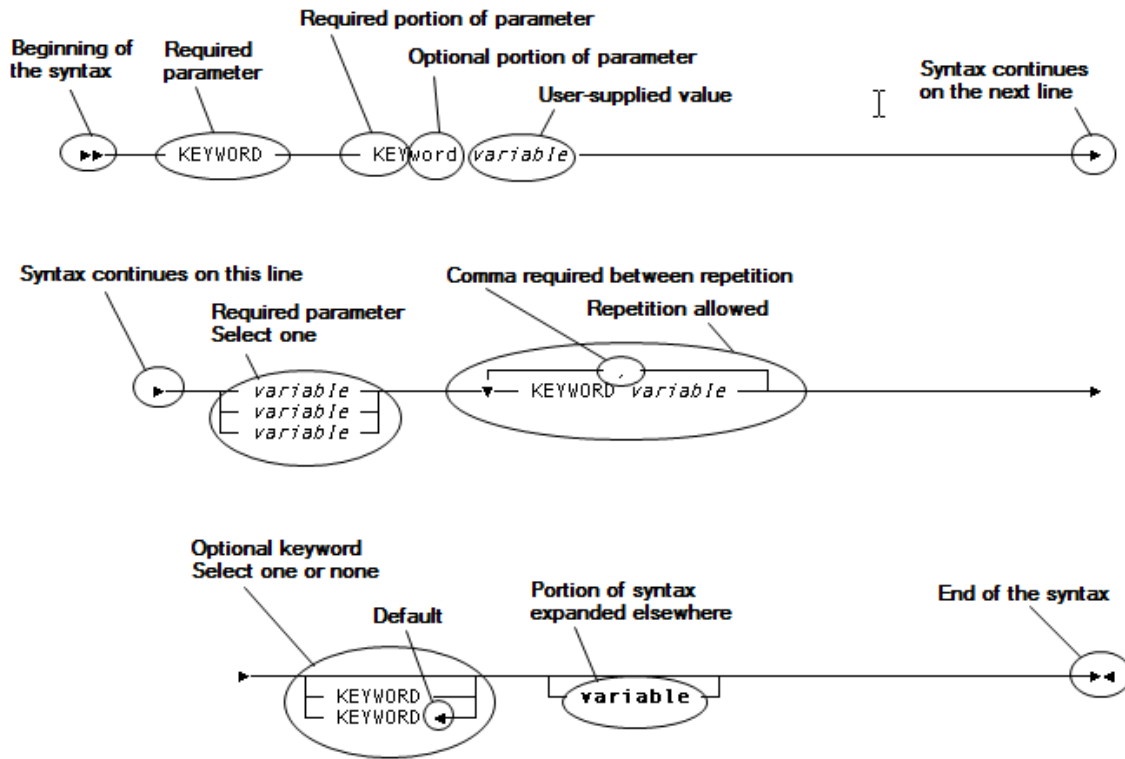
Indicates that you can repeat the parameter or specify more than one parameter.



Indicates that you must enter a comma between repetitions of the parameter.

### Sample Syntax Diagram

The following sample explains how the notation conventions are used:







# Chapter 2: The Automatic System Facility

---

## What is ASF

The **Automatic System Facility (ASF)** is an online, menu-driven facility that enables all users in the corporation to take advantage of database management system capabilities. ASF is a facility of the CA IDMS/DB database management system. ASF is also a component of the Information Center Management System (CA-ICMS), which uses an CA IDMS/DB database. Thus, ASF is a tool for both the **development center** and the **information center**.

## You Define Data Tables

Through ASF, you can accomplish your own data processing tasks, no matter what level of data processing expertise you have. ASF is the interface between you and the database. Through ASF, you define **data tables** by describing the data you want to manage. ASF automatically generates an **application program** to store and access data in the table you define.

## ASF Creates Support Structures

You need only define a data table; ASF creates all the supporting structures needed to use that table. ASF performs the following application development tasks for you:

- Formats screens for displaying/updating data
- Generates programming logic for adding, updating, deleting, and displaying data
- Stores definitions for formatting data

## Create Your Own Database

By using ASF, you can create **private databases** without requiring assistance from the data processing department. Private databases give users direct control over data that is of interest to an individual or to a small group. Users can also have easy access to corporate data that is stored in **production databases**. ASF enables you to easily develop applications that answer many of your data processing needs.

## Use or Add to Existing Database

You can define a table that adds new data elements to the database, or define a table that relates elements already existing in the database. **Relational operations** can be performed through ASF when defining data tables. ASF includes all the functions necessary to define data tables and to use those tables through the online applications created by ASF.

This section contains the following topics:

[Data Tables](#) (see page 19)

[Relational Operations](#) (see page 23)

[ASF in the Corporate Environment](#) (see page 25)

## Data Tables

Data tables are two-dimensional tables that consist of **columns** and **rows**. A column:

- Has entries containing a single type of data
- Is displayed vertically
- Is identified by name

A row:

- Contains one value in each column
- Is displayed horizontally
- Is not named

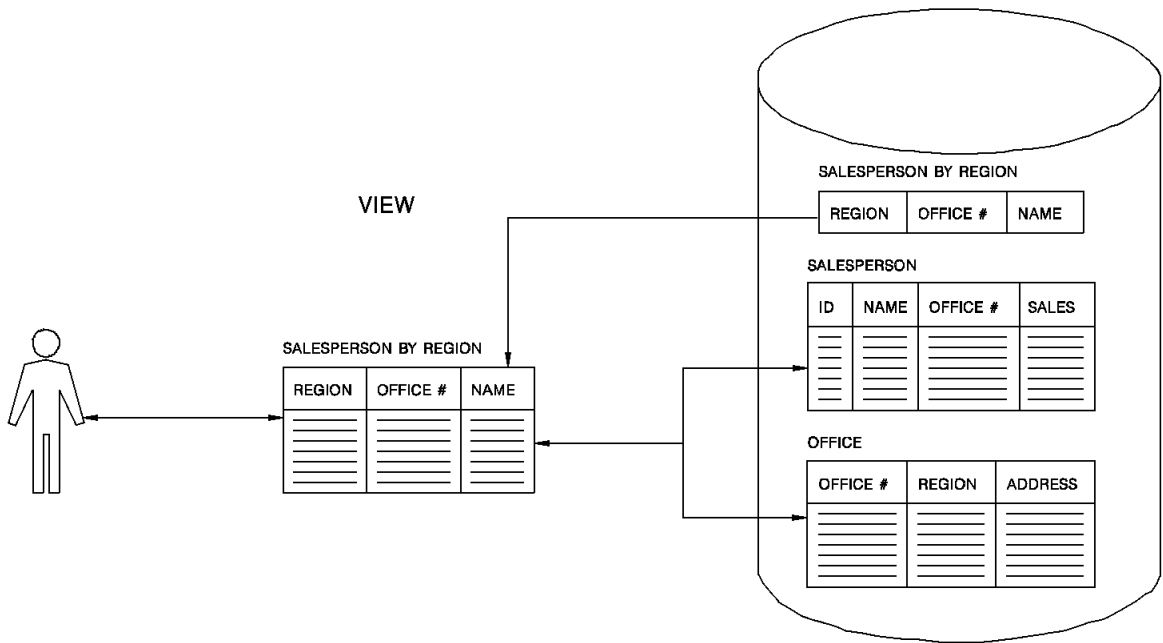
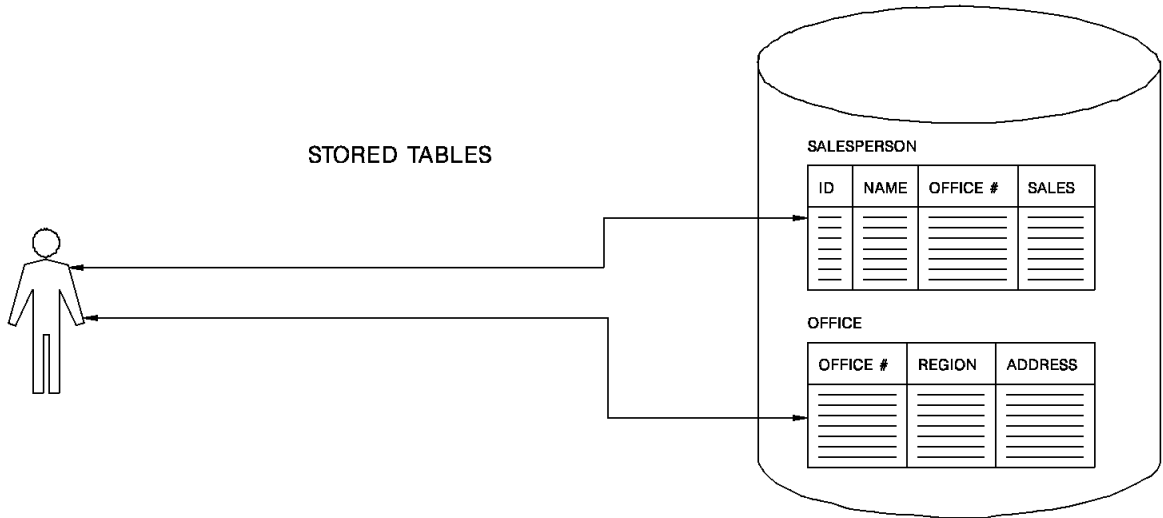
		Column					
		NUMBER	FIRST NAME	LAST NAME	ADDRESS	CITY	STATE
		2453	CATHERINE	HALBURN	57 MAIN ST	BOSTON	MA
		1134	JON	BURKE	23 SOUTH ST	NEWTON	MA
		1469	SUSAN	MAYO	60 COMMONS ST	BOSTON	MA
		0921	LINDA	IOSSA	82 THOMAS AVE	WESTWOOD	MA
		0286	ANGELA	CAMPANIELLO	27 SHELBY ST	WORCESTER	MA
Row	1373	MARGARET	BARBER	86 JONES AVE	DEDHAM	MA	

### Two Kinds of Data Tables

Data tables can be either **stored tables** or **views**:

- **Stored tables** are physically stored in the database and are independent of any other tables.
- **Views** are derived from other tables and records; they do not contain data themselves, but contain a definition of data from the source tables. When a view is accessed, data is retrieved from the source tables in which it is stored.

Stored tables and views are illustrated in the figure below.



---

## Stored Tables

### Defining

You can define stored tables, as an end user, with:

- **ASF**—Name the table columns and define the characteristics of these columns. This description of the data is independent of existing data definitions.

End users and programmers building a prototype can describe the data they want stored in the database.

- **CA OLQ**—Build reports and then save the reports as stored tables. You build reports by selecting columns and rows from data elements that already exist in the database.

End users and programmers can save reports as data tables. By saving the report as a stored table, you can manipulate the data without disturbing the original source data.

- **CA Culprit**—Read input from several types of database and file structures and write the data to a data table.

- **The Information Center Management System (CA-ICMS)**—save stored tables through supported PC software, or minicomputer software.

Personal computer users can load **objects** created through supported PC software. When objects defined as data tables are uploaded from the personal computer, they can be accessed through mainframe products, such as ASF and CA OLQ.

### Using

All tables created using these methods share the following characteristics:

- The underlying structures are created automatically; you do not need to be aware of them.
- You can display the table definition through ASF.
- You can use ASF to create an application that enables you to display, add, change, and erase rows in the table.
- You can manage the table through CA-ICMS.

## Non-SQL Defined Records

Database administrators can code **Data Description Language (DDL)** statements and submit these statements to the schema compiler or the DDDL compiler and create non-SQL defined records.

In this manner, the DBA creates all the underlying structures, taking full advantage of the tuning options available when the schema compiler is used directly.

The definitions of non-SQL defined records cannot be displayed through ASF, nor can these records be categorized through CA-ICMS. The DBA can make the records available to ASF so that they can be used as sources for views.

## Views

Views are data tables that are derived from one or more existing stored tables or non-SQL defined records. By defining a view, you can combine several tables or records and present them as one table. Alternatively, views also enable you to select certain columns and rows from stored tables and records. By defining a view, you can create a table that presents a subset of information in the database.

When a view is accessed, data is retrieved directly from the source tables or records. Thus, a view always presents a current picture of data stored in the database. Source tables and records can be modified through a view if the user has the authority to do so.

### Using

Views can be used to:

- Apply security. A view can be defined for a user that contains only certain columns or rows of data, rather than the entire source table or record.
- Combine several tables or records in order to display them as one relational view.
- Analyze data based on the relationships between tables or records.
- When a user has update authority, modify several source tables or records through one online transaction.

### Creating

Views can be created in two ways:

- **Using ASF**, an end user can define views of data tables created through ASF, CA OLQ, CA Culprit, or the Information Center Management System (CA-ICMS). Additionally, a programmer or DBA can define views of records (defined through the schema compiler) that have been made available to ASF.
- **Using the subschema compiler**, a DBA can define **logical records**; these logical records represent views of records.

### Limits on Use

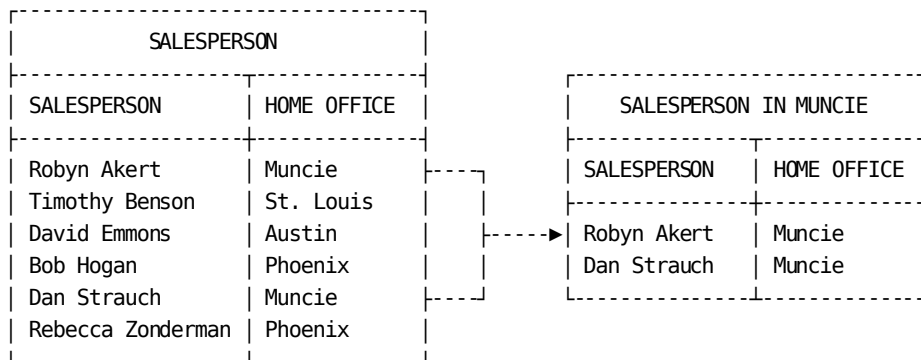
Views created through ASF can be used in ASF as source tables for other views. Logical record views created through the subschema compiler cannot be used in ASF as source tables for other views. Likewise, you cannot derive views from views derived from non-SQL defined records.

## Relational Operations

The three relational operations (**select**, **project**, and **join**) can be used to define and access data tables.

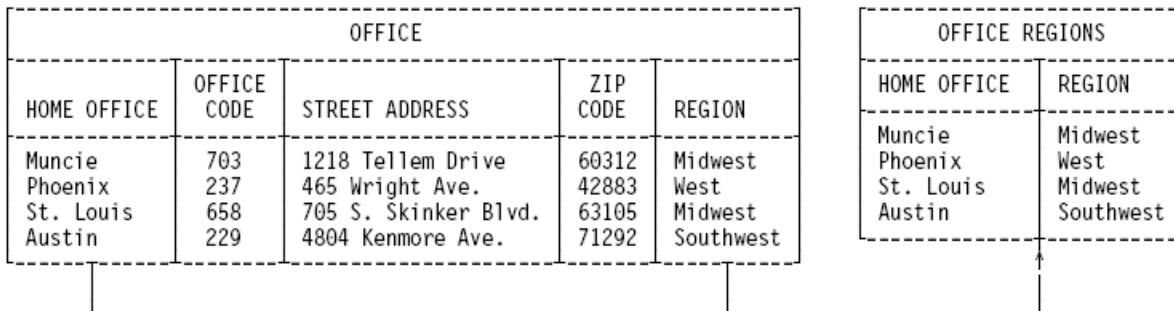
### Select

**Select** enables you to choose the rows to be included in a table.



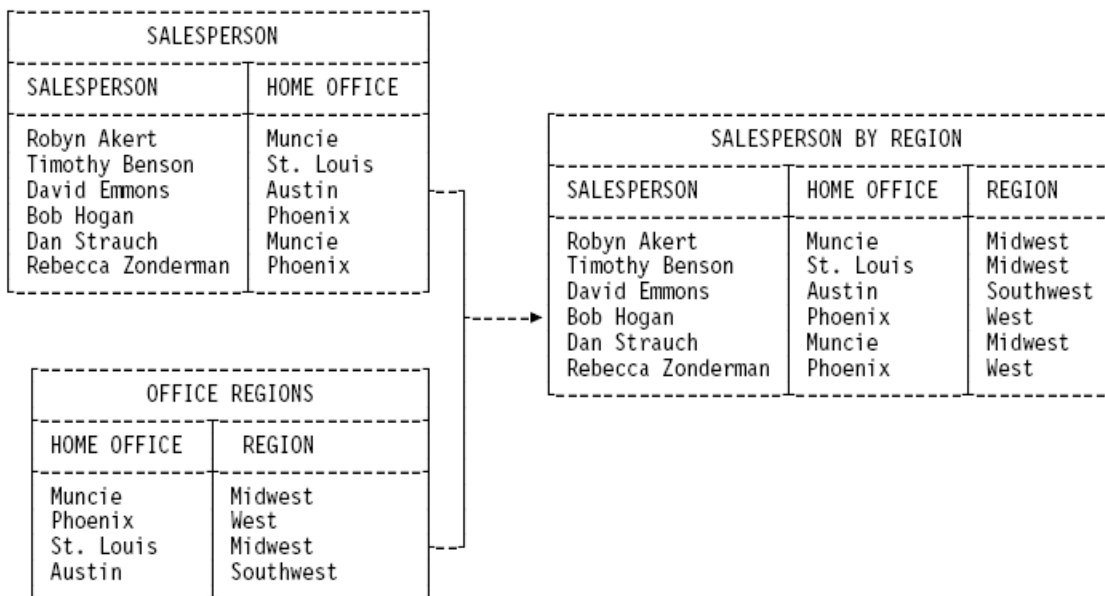
**Project**

**Project** enables you to choose the columns for a table.



**Join**

**Join** enables you to combine two or more tables on the basis of a common column. The tables below are joined using the HOME OFFICE column.



**Perform Relational Operations with ASF**

ASF enables you to perform relational operations easily in various combinations. You can join several tables, project the required columns, and specify detailed criteria for selecting rows of data. Use of the relational operations in ASF is discussed in .



## ASF in the Corporate Environment

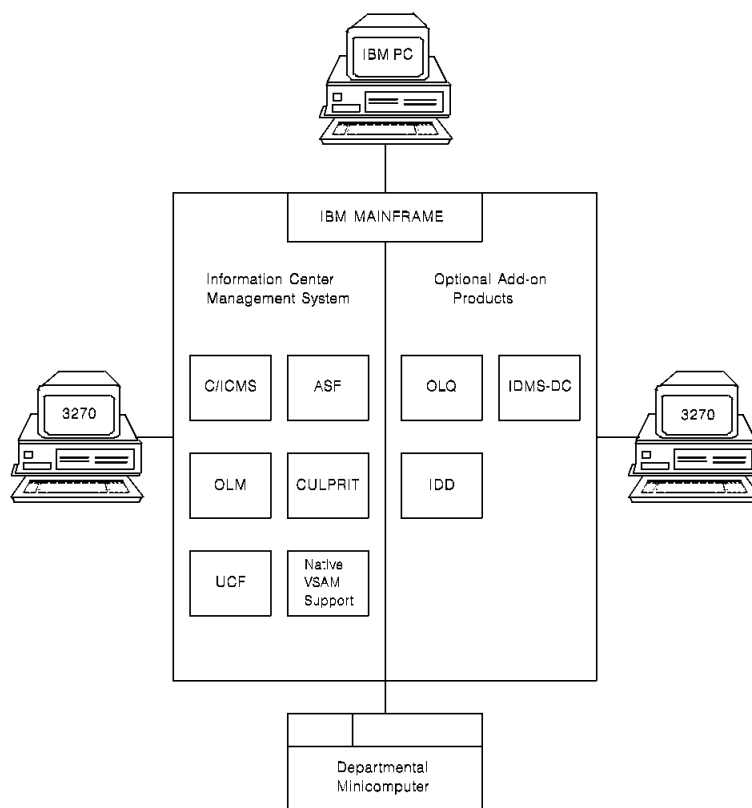
ASF is a component of both the **information center** and the **development center**. The information center centralizes the corporate data resource by linking personal computers, departmental minicomputers, and the corporation's mainframe computers. The development center creates and maintains the corporation's production systems that keep the business running.

### ASF in the Information Center

#### For All Users

The information center addresses the need for end-user computing. The information center products, illustrated in the following figure provide end users with the ability to:

- Develop database applications themselves, without going through the data processing department.
- Share information among personal computers, departmental computers, and mainframe computers.



### Managing the Information Center

The **Information Center Management System (CA-ICMS)** is a mainframe software product for managing the corporate information center. ASF is an integral component of this information center system and is the primary data definition tool for the information center.

### Easy Access to Information

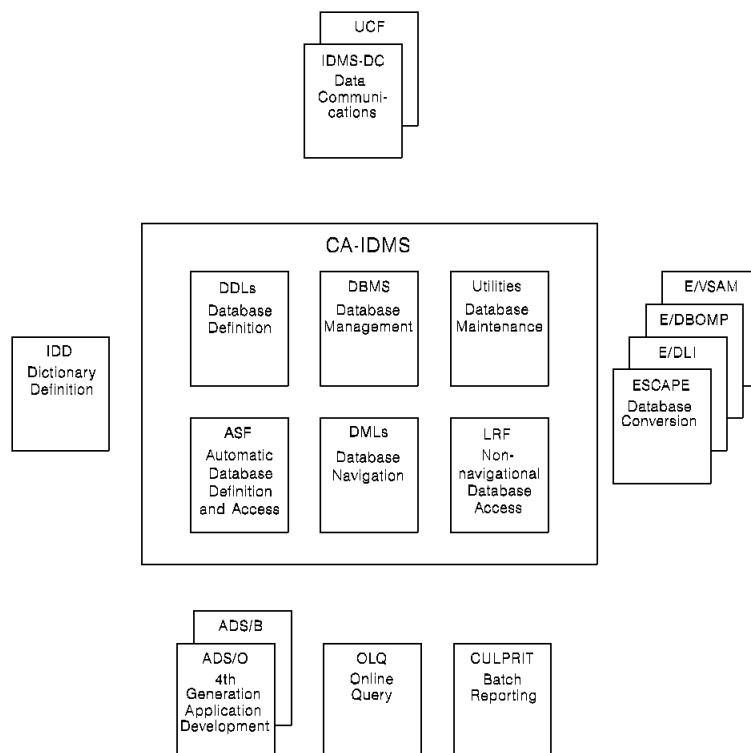
Data is a corporate resource. Managers and other decision-making personnel need access to production data in order to make informed decisions. Through ASF, an information center administrator can define views of production data that can be downloaded to a personal computer, providing managers and other decision-making personnel with immediate access to current data through supported PC software.

### All Users Develop Applications

Information center users can take advantage of the mainframe's computing power by developing database applications through ASF. Data can be stored centrally in the database and shared by all users. CA Culprit and CA OLQ provide users with mainframe reporting capabilities while supported PC software can provide PC users other types of report capabilities.

## ASF in the Development Center

The development center is the corporate data processing department. In the development center, programmers develop management information systems for a production environment. The development center products illustrated in the following figure provide database administrators and programmers with a comprehensive set of database and application development tools.



ASF is a facility of the CA IDMS/DB database management system. ASF has the following functions in the development center:

- **Prototyping tool**—Programmers can use ASF to prototype database applications.
- **Design tool**—Database administrators can design a database online, using ASF to determine relationships between data elements.
- **Generating skeletal program views**—Database administrators can use ASF to automatically generate application program views of the database.

# Chapter 3: Using ASF

---

ASF operates in the CA IDMS/DC and CA IDMS UCF environment. You can access ASF from a 3270-type terminal or from an IBM Personal Computer using 3270 emulation.

This section presents general introductory material on topics that will help you to use ASF effectively.

This section contains the following topics:

[The Structure and Functions of ASF](#) (see page 30)

[ASF Screen Format](#) (see page 37)

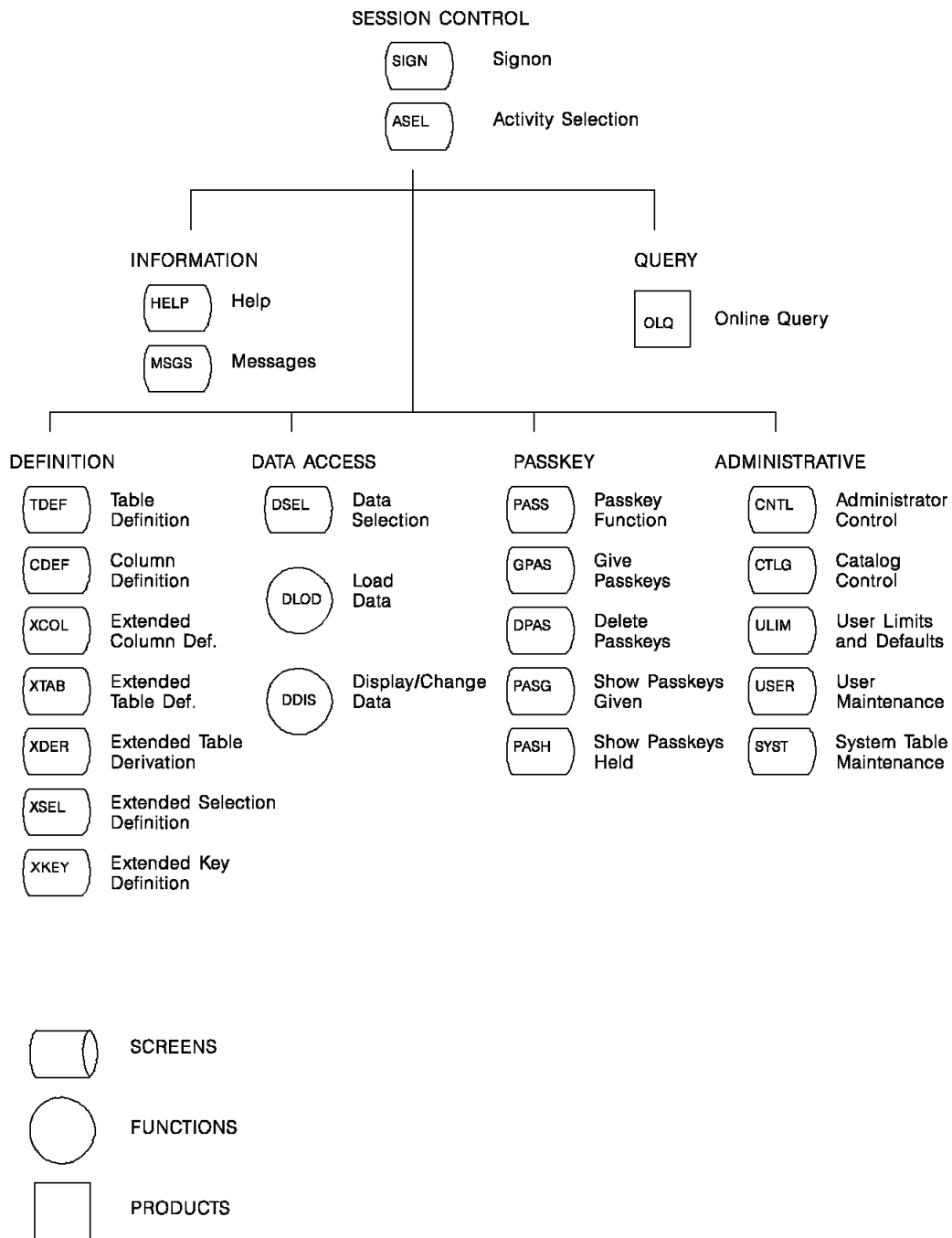
[ASF Screen Management Features](#) (see page 38)

[Conducting an ASF Session](#) (see page 40)

[Selecting Tables](#) (see page 43)

## The Structure and Functions of ASF

ASF consists of a series of menu-driven screens. The screens are grouped into functional categories, as illustrated in the figure below and outlined on the following pages.



The functions and screens in ASF are listed below:

- **Session control** screens enable you to begin and end an ASF session:
  - Signon (SIGN)
  - Activity Selection (ASEL)
- **Information** screens provide information on using ASF:
  - Help (HELP)
  - Message (MSGS)
- **Query** enables you to access OnLine Query (CA OLQ) directly through ASF if CA OLQ is installed at your site.
- **Definition** screens enable you to define data tables (both stored tables and views):
  - Table Definition (TDEF)
  - Column Definition (CDEF)
  - Extended Column Definition (XCOL)
  - Extended Table Definition (XTAB)
  - Extended Table Derivation (XDER)
  - Extended Selection Definition (XSEL)
  - Extended Key Definition (XKEY)
- **Data access** screens enable you to manipulate the data in a table:
  - Data Selection (DSEL)
  - A data display screen accessed through the Load Data (DLOD) and Display/Change Data (DDIS) functions
- **Passkey** screens enable you to give other users access to tables you create:
  - Passkey Function (PASS)
  - Give Passkeys (GPAS)
  - Delete Passkeys (DPAS)
  - Show Passkeys Given (PASG)
  - Show Passkeys Held (PASH)

- **Administrative** screens enable authorized personnel to administer the ASF system:
  - Administrative Control (CNTL)
  - Catalog Control (CTLG)
  - User Limits and Defaults (ULIM)
  - User Maintenance (USER)
  - System Table Maintenance (SYST)
- Row-level security screens enable you to restrict other users' access to rows of data within tables:
  - Display/Change Row-Level Security (RDIS)
  - Load Row-Level Security (RLOD)
  - Row-Level Security (RSEC)
  - Select Row-Level Security (RSEL)

The functional categories and the screens each includes are discussed in more detail below.

## Session Control

An ASF session involves signing on to an ASF system and selecting tables and activities. The two screens used to control an ASF session are:

- **Signon (SIGN)** allows the user to:
  - Sign on to ASF as an authorized user
  - Sign on to the appropriate ASF system
  - Exit ASF if the user does not sign on
- **Activity Selection (ASEL)** allows the user to:
  - Name a new table to be defined
  - Select an existing table
  - Select an activity
  - Exit ASF



## Information

### Online Assistance

ASF provides information screens that help you through an ASF session:

- **Help (HELP)** provides tutorials on the use of the ASF screens. Help text is provided for each ASF screen. The Help facility displays the appropriate text for the screen being used when help is requested.
- **Message (MSG)** provides informational messages related to ASF processing.

## Query

### CA OLQ

The **OnLine Query menu facility** can be accessed directly from ASF. If you have CA OLQ installed, you can select the Query function ([PF13] on the Activity Selection screen) and immediately display the CA OLQ Column Select screen for the table you were using in ASF. You can use CA OLQ to format a report for a table. If you haven't chosen a table in ASF, the Query function takes you to the Data Table Processing screen in CA OLQ.

For more information on using CA OLQ to process a table, see the OnLine Query documentation.

## Definition

Applications are developed through ASF by defining data tables. Data tables can be defined easily by all users through the basic definition screens. ASF also provides extended definition screens for use by data processing professionals.

### Two Basic Definition Screens

The following basic definition screens are always used in defining a data table:

- **Table Definition (TDEF)** allows the user to:
  - Specify the type of table being defined: STORED or VIEW
  - Specify a simple table derivation to define a view
  - GENERATE the table and an application for immediate online use of the table
  - DELETE the table definition, the generated application structures, and, for stored tables, all data stored in the table
- **Column Definition (CDEF)** allows the user to define each column to be contained in the stored table and specify the characteristics of each column.

### Six Extended Definition Screens

The following extended screens are intended primarily for use by data processing professionals in further defining a data table:

- **Extended Column Definition (XCOL)** allows the user to define additional characteristics for a column and to edit the attributes automatically defined for a column.
- **Extended Table Definition (XTAB)** allows the user to override the default space allocation for the table, to specify access restrictions to be included in the ASF-created application, to review information, and if necessary, modify information supplied by ASF about the table.
- **Extended Table Derivation (XDER)** allows the user to perform the following activities:
  - Identify the sources from which a view is being derived
  - Name the owner of the source table or the subschema if the source table was defined through a subschema
  - Specify the conditions on which the source tables are to be joined
- **Extended Selection Definition (XSEL)** allows the user to specify one or more selection criteria that become part of the table definition and are used as edit criteria for loading and/or displaying data.
- **Extended Key Definition (XKEY)** allows the user to define keys used to index the table.
- **Row-Level Security (RSEC)** allows the user to specify the security name for the table and any generic column names.

## Data Access

When you generate a table through ASF, an **application program** that allows you to use the table online can be generated automatically. The application allows you to display, add, modify, and delete rows in the stored table and display and modify rows in a view. As part of the application, a **Data Display screen** is formatted for the table, displaying the columns defined for the table and providing response fields for adding and displaying rows of the table. Though not actually an ASF screen, the Data Display screen for a particular table can be accessed directly from ASF in the following ways:

**Selecting Data to View: Through the Data Selection (DSEL) screen**—ASF provides a run-time selection screen similar to the Extended Selection Definition screen described earlier. The Data Selection screen allows you to specify a value or range of values for one or more columns in the table. Only the rows whose columns match the values specified will be displayed on the Data Display screen.

When you have specified the rows to be selected, the Data Selection screen transfers control directly to the Data Display screen, which displays the first row that matches the selection conditions specified.

The selection criteria do not affect what data is stored in the table, only what data is displayed in this run of the ASF-created application.

**Adding Data : Through the Load Data (DLOD) function**—The Data Display screen displays the column headings so you can enter new data online, one row at a time, and add the new data to the database.

**Row by Row Changes : Through the Display/Change Data (DDIS) function**—The Data Display screen lists the column headings and one row of data and allows you to:

- Modify the table in the database by changing, adding, or deleting data
- Display the entire table, one row at a time

When you access the Data Display screen, you are using the application developed by ASF for the data table. After completing all data access transactions (displaying, adding, changing, and deleting rows) through the table's online application, you return to ASF to continue or to conclude the ASF session.

## Passkey

### Control Access by Other Users

You can allow other users to access the applications you develop through ASF by issuing **passkeys** for your tables. Passkeys provide security for data tables. By granting and removing passkeys, you control what other users can do with the tables you create. For example, if you give a user an ADD passkey for your table, the user can add rows to the table.

You can give a passkey for an individual table, your user ID, which allows the user receiving the passkey to access all of your tables, or to a folder (CA-ICMS users only). Passkeys can be given, deleted, and displayed through the following screens:

- **Passkey Function (PASS)** provides a menu of passkey activities.
- **Give Passkeys (GPAS)** allows you to issue passkeys to another user. The passkeys given determine what the user can do with your table(s).
- **Delete Passkeys (DPAS)** allows you to remove passkeys previously given to another user.
- **Show Passkeys Given (PASG)** allows you to display a list of all the users to whom you have issued passkeys for a particular table, or for your user ID. The list shows which passkeys you have given each user.
- **Show Passkeys Held (PASH)** allows you to display a list of all the passkeys you have received from other users.

## Administrative

### Running ASF

The administrative screens enable personnel authorized as ASF administrators to control and maintain the ASF system. You must have administrative authority in order to access any of these screens. ASF administrators have access to the following screens:

### Control the System

**Administrator Control (CNTL)** allows an ASF administrator to initialize the ASF system, identify database areas that ASF can use to store relational tables, set system-wide defaults, and perform maintenance activities.

### Control Users and Tables

**Catalog Control (CTLG)** allows an ASF administrator to prevent a user from accessing ASF, to give and remove administrative authority, and to maintain the catalog associated with ASF. The catalog is a directory of ASF users and their tables.

### Set User Defaults

**User Limits and Defaults (ULIM)** allows an ASF administrator to set the following defaults for an individual user:

- Database space allocation defaults
- Default models used to create the application
- Global passkey limitations for the ASF system
- Default area name for table data
- Specify limits for unstructured objects

User defaults override the defaults set for the ASF system.

### Add, Delete Users

**User Maintenance (USER)** allows an ASF administrator to add and delete users from the catalog.

### Low Level Security for Users

**System Table Maintenance (SYST)** allows an ASF administrator to initiate row-level security for chosen ASF users.

### Low Level Security for Tables

**Row-Level Security screens** allow you to assign row-level security to stored tables.

- RDIS—Display/Change Row-Level Security
- RLOD—Load Row-Level Security
- RSEL—Select Row-Level Security

## ASF Screen Format

### The First Two Lines

All ASF screens have a similar design. The first two lines of all screens (after the masthead) contain the same fields. These lines and the fields they contain are shown below; explanations of the fields follow.

CA
ASEL1 CA - Automatic System Facility nn.n2 ** Activity Selection3**
4 DC560122 SIGNON COMPLETED, SELECT A TABLES
Screen name

1. Product name and release number
2. Screen title
3. Simulated PF-key field
4. Message display field

- The **first line** contains the product name, release number, and screen title. To the right is a simulated PF-key field; to the left is the screen name field:
  - The **simulated PF-key field** is an untitled field that you can use to simulate PF-keys when actual keys are unavailable. The use of this field is explained in detail under "ASF Screen Management Features" later in this section.
  - The **screen name field** is an untitled field that displays the four-character screen name. This field can be used to move from one screen to another by changing the screen name. Use of this field is explained in detail under "Conducting an ASF Session" later in this section.
- The **second line** contains the system message display field. This untitled field is used by the system to respond to your use of the screen. Messages are displayed to request additional information or action, to indicate that the information supplied has been processed, or to explain why information has not been processed.

The remaining lines on each screen contain menus, information display, prompts, and response fields into which you can enter information. Part III, "Automatic System Facility Screens," contains detailed presentations of each ASF screen.

## ASF Screen Management Features

ASF provides the following features that allow you to access and use the ASF screens efficiently:

- Function key assignments are predefined
- Cursor movement
- Message display and field highlighting
- Default value assignment

### Function Key Assignments Are Predefined

The association between function keys and specific functions is predefined under ASF. You cannot change ASF function key associations. Function key assignments are either global or local.

#### Seven Global Keys

**Global function keys** perform the same functions at any point in an ASF session:

- Press **[Enter]** to submit information to ASF for processing.
- Press **[Clear]** to display the prior screen; press [Clear] from the Activity Selection screen to leave ASF and return to DC/UCF.
- Press **[PA1]** to access the Activity Selection screen from any screen. When on the Activity Selection screen, [PA1] displays the previously listed table directory.
- Press **[PA2]** to refresh the screen. Pressing [PA2] cancels the changes you have just typed on the screen, rewriting the screen as of the last time you pressed [Enter].
- Press **[PF1]** to invoke the Help function of ASF; the appropriate Help screen is displayed for the activity you are performing.
- Press **[PF7]** to scroll backward to the previous page of the currently displayed pageable screen. Pageable screens are indicated by the field Page\_\_Of\_\_.
- Press **[PF8]** to scroll forward to the next page of the currently displayed pageable screen.

### Local Keys Vary with Screens

**Local function keys** perform different functions depending on the screen. The menu on each screen tells you the local function key assignments in effect for that screen.

When you are using a terminal that does not have PF-keys, you can perform PF-key-related functions by entering the appropriate PF-key number in the simulated PF-key field in the upper right corner of the screen. For example, to perform the function associated with [PF2], type **2** in the simulated PF-key field.

On screens with activity selection menus, you can select a PF-key-related function by entering any character in the response field associated with the listed function.

## Cursor Movement

On all ASF screens, you can enter or change information only in response fields; you cannot change information in menu, display, or prompt fields. To ensure quick access to ASF response fields, use the [Tab], [Backtab], [Home], or [Return] key to move the cursor.

When the cursor is at a response field, you can key in new information or replace existing information. Additionally, you can press the space bar or the erase end-of-field key to delete characters across the response field. You can use the cursor control keys to move across the response field without deleting characters.

## Message Display and Field Highlighting

### Message Display

Messages are displayed in the system message display field on the second line of a screen. Messages perform the following functions:

- Request that you supply additional information
- Tell you what action is required next
- Inform you about what has been done with information you entered for processing
- Tell you why information has not been processed
- Report on the status of tables and processing

### Field Highlighting

Highlighting is sometimes used in conjunction with a message to indicate the field on the screen referenced by the message. Highlighting can appear in any information display field on a screen.

## Default Value Assignment

ASF supplies default values in the response fields of some screens. These default values allow you quick access to the information most likely to be used in response fields. Additionally, some default values are supplied as a result of processing performed on a screen.

### Use or Change Defaults

You can decide to use the default values or change them by typing the appropriate value in place of the default value. More information on specific default values is provided in Part III, "Automatic System Facility Screens."

## Conducting an ASF Session

An ASF session begins when you access ASF from DC/UCF; the session ends when you exit from ASF and return control to DC/UCF. During an ASF session, you can use any number and combination of screens to perform various activities.

### A Session Has Three Parts

- Beginning an ASF session
- Moving between the various ASF screens
- Ending an ASF session

## Beginning an ASF session

To begin an ASF session, sign on to DC/UCF. At the ENTER NEXT TASK CODE prompt, type **asf**, as shown below:

```
ENTER NEXT TASK CODE:  
?asf
```

**Note:** The task code prompt and/or the task code to invoke ASF may be different at your installation; check with your ASF administrator if you are unsuccessful in accessing ASF.

Press [Enter]. If your DC/UCF signon is sufficient to access the ASF system, the Activity Selection screen is displayed. If your DC/UCF signon is not sufficient to access the ASF system, the Signon screen is displayed, enabling you to sign on successfully to ASF:

- **Sign on as an authorized ASF user** by entering a valid user name and/or password.
- **Sign on to the correct ASF system** by entering the appropriate catalog.

Press [Enter]. When signon is successful, the Activity Selection screen is displayed.



## Moving Between ASF Screens

When you have accessed an ASF screen, you can move to another screen in any of several ways.

### Use the Menu

Select the function associated with the screen from the activity selection menu:

- Press the appropriate PF-key.
- Enter any character in the response field and press [Enter].
- Type the PF-key number in the upper right corner of the screen and press [Enter].

### Specify a Screen

Enter the screen name in the upper left corner of the screen.

You can always move directly to a screen by entering the screen name in the upper left corner, even if the screen is not listed in the activity selection menu.

```

                                     CA
tdef1 CA - Automatic System Facility nn.n  ** Activity Selection ** 22
DC560002 SELECT NEXT ACTIVITY

User Name: SNELL

_ PF1 - Help           _ PF5 - Select Data       _ PF13 - Query
x3PF2 - Define Table  _ PF7 - Page Backward  _ PF14 - Signon
_ PF3 - Load Data    _ PF8 - Page Forward   _ PA1  - Prior Level
_ PF4 - Display/Change Data _ PF9 - Passkey       _ CLEAR - Leave ASF
    
```

1. Enter screen name in the screen name field.
2. Enter the PF-key number in the simulated PF-key field.
3. Enter a character in the response field or press the PF-key.

### Return

Once you have moved to a screen, you can:

- Press [Clear] to return to the prior screen.
- Press [PA1] to move to the Activity Selection screen.

**Note:** Moving forward to another screen causes changes on the current screen to be processed. Returning to the prior screen by pressing [Clear] **does not** process changes on the current screen.

The following table shows ASF screens and the four-character screen name that can always be used to move directly to the screen.

**Note:** You can review the screen names at any time during an ASF session by accessing the online Help facility (press [PF1]).

Screen title	Screen name
Activity Selection	ASEL
Signon	SIGN
Data Display *	DDIS
* The screen formatted by ASF for the online application created for the table	DLOD
Data Selection	DSEL
Table Definition	TDEF
Column Definition	CDEF
Extended Column Definition	XCOL (valid from CDEF only)
Extended Table Definition	XTAB
Extended Selection Definition	XSEL
Extended Key Definition	XKEY
Extended Table Derivation	XDER
Passkey Function	PASS
Give Passkeys	GPAS
Delete Passkeys	DPAS
Show Passkeys Given	PASG
Show Passkeys Held	PASH
Administrator Control (available with administrator authority only)	CNTL
Catalog Control (available with administrator authority only)	CTLG
User Limits and Defaults (available with administrator authority only)	ULIM
User Maintenance (available with administrator authority only)	USER
System Table Maintenance (available with administrator authority only)	SYST

Screen title	Screen name
Help	HELP
Message	MSGS
Column Select screen of OnLine Query menu facility *	CA OLQ
Display/Change Row-Level Security	RDIS
Load Row-Level Security	RLOD
Row-Level Security	RSEC
Select Row-Level Security	RSEL

\* Other CA OLQ screens can then be accessed from this screen.

## Ending an ASF Session

To end an ASF session, return to the Activity Selection screen by pressing [PA1], then press [Clear] from the Activity Selection screen. This ends the session and returns control to DC/UCF. At this point, you can use another DC/UCF facility, or sign off from the system by entering **bye** at the ENTER NEXT TASK CODE prompt.

## Selecting Tables

Most often, an ASF session involves creating a new table or selecting an existing table for processing.

### Naming a New Table

New tables are created by entering a unique name in the TableName field of the Activity Selection screen. A table name:

- Can be 1 to 32 characters long
- Can include letters, numbers, spaces, and special characters
- Cannot include a period (.), underscore (\_), accent grave (&grave.), or backslash (\), and should not include a cent sign (&cent.)
- Cannot be the same as your user ID
- Cannot begin with spaces

When you enter the name of a new table, you must select DEFINE TABLE as your first activity.

### Listing Tables

Existing tables are listed in the directory at the bottom of the Activity Selection screen. You can browse through the directory to display the names of all your existing tables. Press [PF8] to page the directory forward and [PF7] to page backward.

You can browse through the directory of another user if the user has given you a **passkey** to do so. For information on passkeys, see the chapter "Granting Access to Tables", and the chapter "Passkey Screens". To list another user's directory, enter the user's ID in the Table Owner field, leaving the Table Name field blank.

### Selecting a Table

To select an existing table:

- Type the table name in the Table Name field. If the table is owned by another user, type the user's ID in the Table Owner field.
- Type any character next to the name of a table listed in the directory, whether you are displaying your directory or another user's directory.

When you select an existing table, you can select any activity that you have passkey authorization to perform.

### Tables in Folders

When CA-ICMS is installed, the ASF directory can include **folders** as well as data tables. Data tables can be placed in folders through CA-ICMS. Folders are indicated by three periods (...) after a listed item. The following considerations apply to the use of folders in ASF:

- When you select a folder from the directory, the folder's **access path** is displayed in the Table Name field and the contents of the folder are listed in the directory.
- As you move down the levels of a folder, ASF adds to the access path displayed in the Table Name field. If the access path exceeds 64 characters, the Table Name field displays the minimum path needed to uniquely identify the folder or table selected.

- You can redisplay the prior directory level of a folder by pressing [PA1].
- You can select a table within a folder by finding the table in the directory and selecting it, or by entering an access path that identifies the table in the Table Name field.
- When creating a new table through ASF, you cannot add it to a folder. You can include tables in folders only through CA-ICMS.

### Leaving a Table

Once a table is selected, the directory is no longer displayed. To redisplay the directory and cancel the table selection, press [PA1].

A sample directory is illustrated below. Note that the directory includes two folders, which are indicated by three periods (...) after the folder name.

### Table Directory with Folders

CA
Table Name.:
Table Owner: SNELL
Page: 1 of 1
- APPLICANT
- BUDGET
- EMPLOYEE
- MONTHLY REPORTS...
- PERSONNEL...



# Chapter 4: Creating and Using Stored Tables

---

## Why Use Tables

You create stored tables so you can store, retrieve, and process information. The information may be of concern only to yourself, or it may be used by many people in the corporation.

Before creating stored tables, you must decide what information you need and how you want to use that information.

## An Example

For example, Fran Snell, personnel director of Acme Insurance, wants to keep a record of all the positions in the company, both filled and unfilled. Fran also wants to store information about applicants for positions at Acme Insurance. Fran would like to use the job and applicant information stored in the database to help personnel fill open positions at Acme.

In this section, you will outline an application that meets Fran's data processing needs and create the stored tables that make up the application. This section covers the following topics:

- Outlining a relational application
- Defining a stored table
- Using a table to store and retrieve data
- Stored table summary

This section contains the following topics:

[Outlining an Application](#) (see page 48)

[Defining a Stored Table](#) (see page 51)

[Using a Data Table](#) (see page 59)

[Stored Table Summary](#) (see page 69)

## Outlining an Application

To outline her employment application, Fran:

1. **Determines the required tables** by identifying the major categories of information to be processed.
2. **Defines the columns for each table** that label items of information within each category.
3. **Specifies a unique key for each table:** one column or a combination of columns that uniquely identifies each row of data in the table.
4. **Establishes join columns:** columns common to two tables that are used to combine the tables.

### Step 1: Determining the Required Tables

Fran wants her employment application to do the following:

- Collect, sort, and store job information
- Collect, sort, and store applicant information
- Match qualified applicants with available positions

Fran determines that she needs three tables in her application, as follows:

- A stored table to contain the job information
- A stored table to contain the applicant information
- A view that combines information from the JOB and APPLICANT stored tables

### Step 2: Defining the Columns for Each Table

Now that Fran has established the table categories, she has to decide which items within each category she wants to include. She writes down the column headings for each table and estimates how wide each column should be.

- The JOB table has the following columns:

Column name	Contents
JOB NUMBER	4 characters
DEPARTMENT	20 characters
TITLE	30 characters
SALARY	7 digits



Column name	Contents
CURRENT STATUS (FILLED OR OPEN)	6 characters

- The APPLICANT table has the following columns:

Column name	Contents
NAME	30 characters
STREET	30 characters
CITY	25 characters
STATE	2 characters
ZIP CODE	5 characters
PHONE NUMBER	13 characters, with area code
POSITION APPLIED FOR	30 characters
SALARY REQUIREMENT	7 digits
EVALUATION OF APPLICANT (STRONG, WEAK, OR MEDIUM)	6 characters

- The JOBS AND APPLICANTS view contains columns from the JOB and APPLICANT tables. It also contains an additional column, SALARY DIFFERENTIAL, which is used to calculate the difference between the salary paid for a particular job and the salary required by an applicant for that job.

### 32-Character Limit for Names

Upon reviewing the list of columns, Fran realizes that the EVALUATION OF APPLICANT (STRONG, WEAK, OR MEDIUM) column name in the APPLICANT table is more than the 32 characters allowed by ASF. Fran abbreviates the column name to EVALUATION: STRONG, WEAK, MEDIUM.

## Step 3: Specifying the Unique Keys

### What Is a Unique Key

A column or combination of columns is designated as a **unique key** to uniquely identify each row in a table. By establishing a unique key, you prevent duplicate rows from being stored in a table.

At Acme Insurance, each job in the company is given a unique job number, so Fran designates the `JOB NUMBER` column as the unique key for the `JOB` table.

### A Concatenated Key

Since an applicant can apply for more than one position, and more than one applicant can have the same name, Fran decides that the `NAME` column is not sufficiently unique to serve as the unique key for the `APPLICANT` table. However, the `NAME` column combined with the `PHONE NUMBER` column and `POSITION APPLIED FOR` column is unique and can be the unique key for the `APPLICANT` table. This unique key is also called a **concatenated key** since it is a concatenation of three columns.

### Views do not Have Unique Keys

A view cannot have a unique key since data is not actually stored in a view, but in the source tables from which the view is derived. Therefore, no unique key is specified for the `JOBS AND APPLICANTS` view.

## Step 4: Establishing Join Columns

### What Join Columns Do

Since Fran plans to combine the `JOB` and `APPLICANT` tables to create the `JOBS AND APPLICANTS` view, she must establish a **join column** in each source table. A join column is a column in one table that is repeated in another table so that the two tables can be related.

Fran realizes that the `TITLE` column in the `JOB` table and the `POSITION APPLIED FOR` column in the `APPLICANT` table refer to the same piece of information. Since these columns will contain common values, they can serve as the join columns needed to create the `JOBS AND APPLICANTS` view.

## Defining a Stored Table

To define a stored table through ASF:

- Name the table
- Define the columns
- Generate the table definition and its supporting structures

Sign on to ASF and follow steps 1 through 3 outlined below to create the JOB table for Fran's employment application.

### Step 1: Name the Table

When you first sign on to ASF, the Activity Selection (ASEL) screen is displayed. This screen has three sections:

Section	Purpose
User information	Shows your user ID
Activity selection menu	Lists the ASF functions from which you can choose
Table directory	Shows the tables you own

#### Name the Table

The cursor is positioned at the first entry in the directory, or at the Table Name field if the directory is empty. Position the cursor in the Table Name field. Type **job** in the Table Name field, as shown below:

```

                                     CA
ASEL  CA - Automatic System Facility      ** Activity Selection **
DC560004 SELECT A TABLE

User Name: SNELL

_ PF1 - Help           _ PF5 - Select Data       _ PF13 - Query
_ PF2 - Define Table  _ PF7 - Page Backward    _ PF14 - Signon
_ PF3 - Load Data     _ PF8 - Page Forward     _ PA1  - Prior Level
_ PF4 - Display/Change Data _ PF9 - Passkey         _ CLEAR - Leave ASF

Table Name.: job
Table Owner: SNELL

                                     Page: 1 of 1

```

### Select Define Table Function

After you name the table, select the Define Table function from the activity selection menu in one of the following ways:

- Press [PF2].
- Type **2** in the simulated PF-key field and press [Enter].
- Enter any character (except a blank or underscore character) in the response field for the Define Table function.
- Type **tdef** in the screen name field.

**Note:** In subsequent steps, the PF-key associated with a function is shown in parentheses. You can select any function by pressing the PF-key shown, entering the PF-key number in the simulated PF-key field, typing a character in the response field for the listed function, or entering the appropriate screen name in the screen name field. In any sample screens where the next function is indicated, the function is selected from the menu by typing **x** in the response field, or the appropriate screen name is entered in the screen name field.

### Table Definition Screen

The Table Definition (TDEF) screen is now displayed. Notice that the table name JOB is in the Table Name field. Your user ID is in the Table Owner field.

The View/Stored field below the Table Owner field contains the default value STORED.

The Defn Number (definition number) field is blank. The Status field below the Defn Number field contains the value NEW, indicating that this is a new table that has not yet been defined.

### Select Define Columns Function

Now you are ready to define the columns of the JOB table. Select the Define Columns function ([PF2]) from the activity selection menu, as illustrated below.

```

TDEF  CA - Automatic System Facility          CA          ** Table Definition **
DC560005 MODIFY AND/OR SELECT NEXT ACTIVITY

      _ PF1 - Help                          _ PF4 - Extended Table Definition
      x1PF2 - Define Columns                 _ PF5 - Delete Table Definition
      _ PF3 - Generate                       _ PF6 - Message Screen

Table Name.: JOB
Table Owner: SNELL
View/Stored: STORED2
Comments...:

Defn Number:
Status.....: NEW3

```

1. Select the DEFINE COLUMNS function.
2. The JOB table is being defined as a stored table.
3. The table status is NEW since the JOB table is not yet defined.

## Step 2: Defining the Columns

### The Column Definition Screen

The Column Definition (CDEF) screen is now displayed. Notice that the table name JOB is in the Table Name field.

The cursor is positioned below the Column Name field heading. The first column name to be defined for the JOB table is JOB NUMBER. Type **job number** in the Column Name response field.

Move the cursor to the Width field. The column width is the number of characters the column will accept. Assign the JOB NUMBER column a width of four by typing **4** in the Width field.

### Column Data Type

You can optionally specify the type of entry the column can accept. The entry types can be TEXT, NUMERIC, or CURRENCY. Since TEXT is the default, and the JOB NUMBER column can accept alphanumeric values, you do not need to complete the Type field. Press [Return].

```

CA
CDEF CA - Automatic System Facility      ** Column Definition **
DC560306 NO COLUMNS DEFINED FOR TABLE
Table Name: JOB

  _ PF1 - Help                          _ PF3 - Extended Selection Definition
  _ PF2 - Extended Column Definition    _ PF4 - Extended Key Definition

Page: 1 of 1

      Column Name      Width  Type  Unique  Display  Ext
job number1           42
_3
  
```

1. Enter JOB NUMBER to name the column.
2. Give JOB NUMBER a width of 4.
3. Position the cursor to define the next column.

Enter the remaining columns, shown below, for the JOB table:

Column name	Width	Type
DEPARTMENT	20	
TITLE	30	
SALARY	7	C
CURRENT STATUS (FILLED OR OPEN)	6	

### Define the Columns

To define the columns:

1. Type **department** (the first column name listed above) below the JOB NUMBER column you have already entered.
2. Position the cursor in the Width field.
3. Type the width shown for the column.
4. Move the cursor to the next Column Name field position.
5. Repeat until all the column names and widths have been entered.
6. For the SALARY column, also tab to the Type field and type **c** to indicate that the entry type is CURRENCY.

The CURRENCY column type causes ASF to provide two decimal positions for a numeric column, enabling you to enter dollar and cent amounts as column values.

```

CA
CDEF CA - Automatic System Facility      ** Column Definition **
DC560306 NO COLUMNS DEFINED FOR TABLE
Table Name: JOB

  _ PF1 - Help                          _ PF3 - Extended Selection Definition
  _ PF2 - Extended Column Definition    _ PF4 - Extended Key Definition

Page: 1 of 1

      Column Name      Width  Type  Unique  Display  Ext
job number             4
department             20
title                  30
salary                 7      c
current status (filled or open) 6

```

**ASF Updates the Screen**

When you have entered all the columns, press [Enter]. ASF rewrites the screen and fills in appropriate default values, as follows:

- The Type field has a default value of TEXT for each column, except the SALARY column, for which it has the value CURRENCY.
- The Unique Key field displays an underscore (\_) for each column, since you did not specify a unique key.
- The Display Seq field has a display sequence number for each column, numbered consecutively.
- The Ext Screen field displays an underscore (\_) for each column, which can be typed over to access the extended definition for a particular column.

Your screen should look like the Column Definition screen below.

```

CA
CDEF CA - Automatic System Facility      ** Column Definition **
DC560307 COLUMNS UPDATED
Table Name: JOB

  _ PF1 - Help                          _ PF3 - Extended Selection Definition
  _ PF2 - Extended Column Definition    _ PF4 - Extended Key Definition

Page: 1 of 1

      Column Name      Width  Type  Unique  Display  Ext
      JOB NUMBER      4  TEXT  _       1       -
      DEPARTMENT      20  TEXT  _       2       -
      TITLE            30  TEXT  _       3       -
      SALARY           7  CURRENCY  _       4       -
      CURRENT STATUS (FILLED OR OPEN)  6  TEXT  _       5       -
    
```

**Modify Column Definitions**

Once ASF accepts the column definitions, you can add columns and/or modify columns. For this example, make the following changes to the column definitions you have entered:

- Tab to the Unique Key field for the JOB NUMBER column. Type x to designate the JOB NUMBER column as the unique key for the JOB table. ASF will create an index on JOB NUMBER, thus preventing any duplicate job numbers from being stored in the table.
- Change the Display Seq field for each column by numbering the display sequence 2, 4, 6, 8, 10. This causes ASF to leave a blank line between each column when the table is displayed on the Data Display screen.

**Note:** These characteristics could have been entered when you defined the columns initially, but are being entered as modifications for the purpose of this example.



```

CA
CDEF CA - Automatic System Facility      ** Column Definition **
DC560307 COLUMNS UPDATED
Table Name: JOB

  _ PF1 - Help                          _ PF3 - Extended Selection Definition
  _ PF2 - Extended Column Definition    _ PF4 - Extended Key Definition

Page: 1 of 1

Column Name      Width  Type      Unique  Display  Ext
                Width  Type      Key     Seq     Screen
JOB NUMBER       4  TEXT      -       1       -
DEPARTMENT      20  TEXT      -       2       -
TITLE           30  TEXT      -       3       -
SALARY          7  CURRENCY -       4       -
CURRENT STATUS (FILLED OR OPEN) 6  TEXT      -       5       -

```

1. Type X to define JOB NUMBER as the unique key.
2. Renumber the columns' display sequence.

Press [Enter] to record the changes with ASF. When ASF accepts the changes, press [Clear] to return to the Table Definition screen.

#### The Table Definition Number

Notice on the Table Definition screen (now displayed) that a definition number has been assigned to the table and is displayed in the Defn Number field. The table definition number is a unique number assigned to the table by ASF. The Status field displays the message UNGENERATED. This indicates that you have successfully defined the stored table and must now generate it.

## Step 3: Generating the Table

When you generate a table through ASF, the table definition is completed and an application that enables you to access the table through ASF is created.

#### Select the Generate Function

To generate the JOB table, select the Generate function ([PF3]) from the activity selection menu on the Table Definition screen.

**Note:** There is no screen name associated with the Generate function. This function must be selected from the Table Definition screen.

### ASF Generates the Table

The Table Definition screen remains displayed while ASF:

- Completes the table definition
- Generates programming logic for accessing the table
- Formats a screen for displaying the table
- Creates an application program for adding, modifying, and deleting data in the table
- Documents the table definition and application components in the data dictionary

When the generation is complete, the Status field changes from UNGENERATED to GENERATED, and the message TABLE GENERATED SUCCESSFULLY is displayed.

```
CA
TDEF CA - Automatic System Facility          ** Table Definition **
DC560005 MODIFY AND/OR SELECT NEXT ACTIVITY

  _ PF1 - Help                               PF4 - Extended Table Definition
  _ PF2 - Define Columns                     PF5 - Delete Table Definition
  x PF3 - Generate                           PF6 - Message Screen

Table Name.: JOB
Table Owner: SNELL                          Defn Number: 127
View/Stored: STORED                         Status.....: UNGENERATED
Comments...:

CA
TDEF CA - Automatic System Facility          ** Table Definition **
DC560208 TABLE GENERATED SUCCESSFULLY1

  _ PF1 - Help                               PF4 - Extended Table Definition
  _ PF2 - Define Columns                     PF5 - Delete Table Definition
  _ PF3 - Generate                           PF6 - Message Screen

Table Name.: JOB
Table Owner: SNELL                          Defn Number: 127
View/Stored: STORED                         Status.....: GENERATED2
Comments...:
```

1. The message tells you when the table is generated.
2. The table status changes to GENERATED.

### You Can Use the Table

Now that the table is generated, you can use the application that was created for it to load data into the table.

Press [PA1] to return to the Activity Selection screen.

## Using a Data Table

The purpose of defining a table is to use the table as a vehicle for storing data in the database. The Automatic System Facility not only provides a convenient means of defining data tables, but it also automatically provides online applications for adding, retrieving, and updating the rows of those tables. The application you created when you generated the JOB table enables you to perform the following activities through ASF:

- Loading data into the table
- Displaying data
- Changing data
- Deleting data
- Selecting data for display
- Displaying a table through OnLine Query

### Define Columns, Use Rows

When you **defined** the JOB table, you were concerned with the **column headings**. When you **use** the JOB table, you will be concerned with the column entries, or **rows**, which contain the actual data you are storing in the database. When you load data, you assign values to the columns. Each row gives you all the information about a particular job.

Now that you have defined and generated a stored table, you can load data into the table and manipulate that data. Follow steps 1 through 6 outlined below to load and manipulate the rows of the JOB table.

## Step 1: Loading Data

### Select Load Data Function

The Activity Selection screen is displayed. The JOB table has been selected; JOB is displayed in the Table Name field. Select the Load Data ([PF3]) function from the menu on the Activity Selection screen.

### The Data Display Screen

The Data Display screen is displayed. This is the screen that was formatted for the table as part of the application you generated.

The column names you defined for the table are listed down the left-hand side of the screen. The cursor is positioned in the response field for the first column.

Each row of the table is loaded individually. Type the column entries for the first row as follows:

Columns	Column Entries
JOB NUMBER	a261
DEPARTMENT	underwriting
TITLE	assistant underwriter
SALARY	13500
CURRENT STATUS (FILLED OR OPEN)	filled

### Load the Row

When you have entered all the column entries for the row, load it into the JOB table by pressing [PF1]. When the row is loaded, the message display field displays the message ROW HAS BEEN ADDED.

CA

PF1 = ADD; PF2 = CHANGE; PF3 = DELETE; ENTER = NEXT; CLEAR = EXIT

JOB

JOB NUMBER

DEPARTMENT

TITLE

SALARY .00

CURRENT STATUS (FILLED OR OPEN)

```

                                CA
PF1 = ADD; PF2 = CHANGE; PF3 = DELETE; ENTER = NEXT; CLEAR = EXIT
JOB
JOB NUMBER                a261
DEPARTMENT                underwriting
TITLE                    assistant underwriter
SALARY                    13500
CURRENT STATUS (FILLED OR OPEN) filled

```

Press [Enter] to clear the data from the screen. The column names remain fixed so you can enter the next row. Enter and load all the rows for the JOB table shown below:

JOB NUMBER	DEPARTMENT	TITLE	SALARY	CURRENT STATUS (FILLED OR OPEN)
C417	CLAIMS	CLAIMS ADJUSTER	17000.00	OPEN
O198	OFFICE SERVICES	MAIL CLERK	7000.00	OPEN
B359	BILLING	BILLING CLERK	10000.00	OPEN

#### Load Rows One at a Time

Remember to load each row individually by selecting the Add function ([PF1]) after typing in the data. Wait for the ROW HAS BEEN ADDED message before pressing [Enter].

**Note:** You do not have to clear the data from the screen before entering the next row; you can type over the data already loaded. This can save keystrokes when some entries are the same in several rows. Be sure to erase any data from the previous row that does not belong in the current row.

#### Return

When you have loaded all the rows for the JOB table, press [Clear] to return to the Activity Selection screen.

## Step 2: Displaying Data

### Select Display/Change Data

The Activity Selection screen is displayed. Select the Display/Change Data function ([PF4]) from the activity selection menu.

### Individual Rows

The Data Display screen is displayed. The columns of the JOB table are listed down the left-hand side of the screen. The first row in the table is displayed beside the column names.

Press [Enter]. The first row is replaced by the next. Notice that the column names remain fixed on the screen; only the data is rewritten.

To examine all the rows of the JOB table, continue to press [Enter]. Each row is displayed individually. Notice that the rows have been sorted by job number. The unique key is used by ASF to sort the rows of the table.

### After the Last Row

When you have displayed the last row, the data is cleared from the screen and the message ALL ROWS HAVE BEEN DISPLAYED appears in the message display field. Press [Enter] one more time to redisplay the first row.

## Step 3: Changing Data

### Check the Data

Review the first row once again. Check to see that it was entered correctly. If there are any keying errors, move the cursor to the entry you want to correct and edit the data as necessary.

For this sample application, move the cursor to the Current Status field. Change the job status from **FILLED** to **OPEN**.

### Registering Changes

When the row is displayed on the screen the way you want it stored in the database, you must modify the data in the database by selecting the Change function ([PF2]).

The message ROW HAS BEEN CHANGED displays in the message display field.

**Continue to End**

Press [Enter] to display the next row. Make any necessary modifications to this row. Press [PF2] to implement the change in the database. Wait for the ROW HAS BEEN CHANGED message to display.

Press [Enter] to display the next row. Edit this and the remaining rows, as needed. Remember to press [PF2] to modify each edited row in the database.

## Step 4: Deleting Data

Display the first row once again. For this sample application, delete this row by selecting the Delete function ([PF3]).

The message ROW HAS BEEN DELETED displays in the message display field. This message indicates that the data has been erased from the database. Press [Clear] to return to the Activity Selection screen.

## Step 5: Selecting Data

The Activity Selection screen is displayed. Select the Select Data function ([PF5]) from the activity selection menu.

**The Data Selection Screen**

The Data Selection screen appears. This screen enables you to specify selection criteria. ASF will display only the rows that match the criteria you specify.

**Columns Have ID Numbers**

The column names of the table are listed below the Column Name field heading. Each column has been assigned an identification number, which is listed below the ID field heading to the left of the Column Name field. You can use the ID numbers in place of column names when specifying selection criteria.

In this example, you want to select the rows from the JOB table that have a current status of OPEN.

### **Select Open Jobs**

The cursor is at the first position under the Selection Criteria heading. Press [Return] to move the cursor to the CURRENT STATUS column name. Type 'open', enclosed in single quotation marks, in this field.

### **EQ is the Default Selector**

You do not need to include the conditional operator EQ in the selection criteria because it is assumed by ASF. You can also select rows on other conditions, such as less than (LT), greater than (GT), or not equal (NE). When using other conditions, you must specify the conditional operator. For example, you can enter **ne 'open'** to select the rows where the CURRENT STATUS column entry is not equal to OPEN.

### **ASF Saves the Criteria**

You have just specified that you want to display the rows of the JOB table in which the CURRENT STATUS column entry is OPEN. Press [Enter].

ASF saves the selection criteria you have entered and displays the message SELECTION MODIFIED, HIT ENTER TO PROCEED TO DATA SELECTION. Press [Enter] again.

### **ASF Displays Open Jobs**

The Data Display screen is displayed automatically. This screen displays the columns of the JOB table and the first row that matches the selection criteria you specified. Notice that the Current Status field has a value of OPEN.



**Note:** If you make an error when entering the selection criteria, you remain on the Data Selection screen. The field names to which the erroneous selection criteria apply are highlighted. The message display line instructs you to access the Message screen, which lists all the error conditions.

```
CA
DSEL CA - Automatic System Facility      ** Data Selection **
DC560603 ENTER SELECTION CRITERIA
Table Name: JOB
      _ PF1 - Help      _ PF2 - Message Screen      Page: 1 of 1

  ID      Column Name      Selection Criteria
0001 JOB NUMBER
0002 DEPARTMENT
0003 TITLE
0004 SALARY
0005 CURRENT STATUS (FILLED OR OPEN) 'open'
```

Additional Selection Criteria

```
CA
DSEL CA - Automatic System Facility      ** Data Selection **
DC560604 SELECTION MODIFIED, HIT ENTER TO PROCEED TO DATA SELECTION
Table Name: JOB
      _ PF1 - Help      _ PF2 - Message Screen      Page: 1 of 1

  ID      Column Name      Selection Criteria
0001 JOB NUMBER
0002 DEPARTMENT
0003 TITLE
0004 SALARY
0005 CURRENT STATUS (FILLED OR OPEN) 'OPEN'
```

Additional Selection Criteria

CA	
PF1 = ADD; PF2 = CHANGE; PF3 = DELETE; ENTER = NEXT; CLEAR = EXIT	
JOB	
JOB NUMBER	B359
DEPARTMENT	BILLING
TITLE	BILLING CLERK
SALARY	10000.00
CURRENT STATUS (FILLED OR OPEN)	OPEN

Press [Enter] to display the next selected row. Continue to press [Enter] until all the rows you selected have been displayed.

**After using Data**

Press [Clear] to return to the Data Selection screen. Press [Clear] again to return to the Activity Selection screen.

## Step 6: Displaying a Table Through CA OLQ

**Display or Print Tables**

If you have OnLine Query, and you are defined as a user of CA OLQ, you can use the Query function of ASF. The Query function enables you to display a data table in table format and to print a table as a report.

**Select QUERY**

Select QUERY ([PF13]) from the Activity Selection screen. The Column Select screen of CA OLQ is displayed, listing the columns of the JOB table.

**Note:** The column names listed by CA OLQ are in the stored format used internally by ASF. In the stored format, spaces and special characters are replaced with hyphens (-).

CA-OLQ		*** COLUMN SELECT ***	
->		PAGE	1 OF 1
124000	Select columns, specify selection criteria and press ENTER key		
COLUMNS CURRENTLY SELECTED:	5	SELECTION CRITERIA	
JOB			
X 03 JOB-NUMBER		*	
X 03 DEPARTMENT-0002			
X 03 TITLE			
X 03 SALARY			
X 03 CURRENT-STATUS-FILLED-OR-OPEN			

### Select Columns and Rows

This screen enables you to select the columns of the table you want to display, and to specify criteria for selecting rows of the table. Depending on the default in effect at your site, either all the columns in the table are selected (marked by X) or none of the columns are selected (marked by D). For this example, select all the columns in the table.

Press [Enter]. The Retrieval Completed screen is displayed. This screen lists the activities you can perform on the table. By default, the Display Report function is selected.

```

CA-OLQ                                     *** Retrieval Completed ***
->
130000 Select activity and press the ENTER key

Number of whole rows. . . . .           4
Total number of records read. . . . .  4
Total number of records selected. . . .  4
Number of data errors . . . . .         0

Select                                     Command/
Option   ---> Display/Format Activity <---  Screen Name

   X    Display report                     DISplay
   -    Save report                         SAVe
   -    Choose the sort sequence of report  SORT
   -    Change column headers               HEAdEr
   -    Change page header and footer       PAGe HEAdEr
   -    Change display format of data ($,commas) PICTure
   -    Format columns (Alignment, sparse)   EDIT
   -    Specify summary computations (Totals) GROUp BY
   -    Send the report to a printer        PRInt

1=HELP           3=QUIT           4=MESSAGE           6=MENU

```

Press [Enter]. The Display Report screen is displayed. This screen shows the columns and rows of the JOB table.

```

CA-OLQ                                     *** Display Report ***
->                                         Page      1 Line    1
104009 DISPLAY RIGHT to see more report columns
125004 Press ENTER for DISPLAY/FORMAT ACTIVITY selections
                                         JOB REPORT
                                         mm/dd/yy

JOB NUMBER      DEPARTMENT      TITLE      SALARY
-----
B359            BILLING            BILLING CLERK      10000.00
C417            CLAIMS            CLAIMS ADJUSTER    17000.00
0198            OFFICE SERVICES   MAIL CLERK         7000.00
END OF REPORT
CA-OLQ                                     *** Display Report ***
->                                         Page      1 Line    1
125004 Press ENTER for DISPLAY/FORMAT ACTIVITY selections
                                         JOB REPORT
                                         mm/dd/yy

TITLE      SALARY  CURRENT STATUS (FILLED OR OPEN)
-----
ASSISTANT UNDERWRITER  13500.00  FILLED
BILLING CLERK          10000.00  OPEN
CLAIMS ADJUSTER        17000.00  OPEN
MAIL CLERK             7000.00   OPEN
END OF REPORT

- 1 -
1=HELP      3=QUIT      4=MESSAGE   6=MENU      10=LEFT     11=RIGHT
    
```

To display the entire table, press the appropriate PF-keys to page horizontally, or type the commands DISPLAY RIGHT and DISPLAY LEFT on the command line.

### Returning from CA OLQ

When you have displayed the entire table, press [Clear].

The Menu screen appears, listing all the activities you can perform with CA OLQ. For more information about CA OLQ activities, see the **CA OLQ Online Query for CA IDMS User Guide**

Press [Clear] from the CA OLQ Menu screen to return to the Activity Selection screen of ASF.

## Stored Table Summary

The following list summarizes the steps involved in defining and using stored tables through the Automatic System Facility:

1. **Name the table** by entering the table name in the Table Name field on the Activity Selection screen and selecting the Define Table function ([PF2]).
2. **Define the columns** by selecting the Define Columns function ([PF2]) from the Table Definition screen. On the Column Definition screen, enter the column names and widths. Optionally, enter an entry type of NUMERIC or CURRENCY to override the default type of TEXT. Press [Enter] to register the columns with ASF. ASF assigns default values to the Type and Display Seq fields.
3. **Modify the column definitions** as needed and press [Enter] to record the changes with ASF.
4. **Generate the table** by returning to the Table Definition screen from the Column Definition screen (press [Clear]) and selecting the Generate function ([PF3]).
5. **Load data** into the table by returning to the Activity Selection screen (press [PA1]) and selecting the Load Data function ([PF3]). Enter the rows on the Data Display screen, loading each one individually by pressing [PF1].
6. **Display, change, and remove rows** by returning to the Activity Selection screen (press [PA1]) and selecting the Display/Change Data function ([PF4]).
7. **Select rows** for display by returning to the Activity Selection screen (press [PA1]) and selecting the Select Data function ([PF5]). Enter your selection criteria and press [Enter]. Press [Enter] again to display the selected data on the Data Display screen.
8. **Return to the Activity Selection screen** from the Data Display screen by pressing [PA1] once to return to the Data Selection screen, and [PA1] a second time to return to the Activity Selection screen.
9. **Display the table through CA OLQ** by selecting the Query function ([PF13]) from the Activity Selection screen if you have OnLine Query.

**Create APPLICANT Table**

To review stored tables, create the APPLICANT stored table for Fran Snell's employment application. The APPLICANT stored table contains information about applicants for positions at Acme Insurance, as shown below.

NAME	STREET	CITY	STATE	ZIP CODE	PHONE NUMBER
HARRISON NELSON	5 MAIN STREET	WATERTOWN	MA	02172	(617) 924-4381
FRANK HOGAN	10 MAPLE STREET	ARLINGTON	MA	02174	(617) 643-5555
PETER ROSS	231 OAK STREET	HARTFORD	CT	06107	(202) 521-1212
PATRICIA NORRIS	95 ACORN LANE	WESTBURY	NY	11568	(516) 333-3343
LESLIE ANN SMITH	17 ALLEN ROAD	DERRY	NH	03038	(603) 434-6120
BRENDAN BERGWINKLE	70 BREEZY POINT ROAD	YARMOUTH	MA	02673	(508) 398-2476
JOHN BURCHILL	77 MASSACHUSETTS AVENUE	CAMBRIDGE	MA	02139	(617) 253-2341
MARY ELLEN BUSHNELL	15 HEDGE ROW	CONCORD	MA	01742	(508) 369-4576
WALTER COX	29 HALL AVENUE	NASHUA	NH	03060	(603) 344-0983
DONALD CAMERON	33 MIDDLE STREET APT. 2	WALTHAM	MA	02154	(617) 647-1987
JANET CRIMLISK	145 ORCHARD STREET	SOMERVILLE	MA	02144	(617) 625-1450
TIMOTHY MURRAY	68 BUTLER STREET	DORCHESTER	MA	02124	(617) 653-6423
MARTIN MCGOVERN	113 VAN BUREN STREET	TAUNTON	MA	02780	(508) 238-9503
PATRICIA ROAN	23 JANE TERRACE	BROCKTON	MA	02421	(508) 586-1650

POSITION APPLIED FOR	SALARY REQUIREMENT	EVALUATION: STRONG, WEAK, MEDIUM
ASSISTANT UNDERWRITER	15000.00	MEDIUM
ASSISTANT UNDERWRITER	12000.00	STRONG
BILLING CLERK	7500.00	WEAK
BILLING CLERK	12000.00	MEDIUM
CLAIMS ADJUSTER	18000.00	STRONG
BILLING CLERK	12000.00	MEDIUM
CLAIMS ADJUSTER	20000.00	WEAK
CLAIMS ADJUSTER	16500.00	STRONG
CUSTOMER SERVICE REP	13000.00	STRONG
CUSTOMER SERVICE REP	12000.00	MEDIUM
CUSTOMER SERVICE REP	11500.00	STRONG
MAIL CLERK	6000.00	STRONG
MAIL CLERK	7000.00	WEAK
MAIL CLERK	7500.00	MEDIUM

Define the APPLICANT table by naming it on the Activity Selection screen and defining the following columns on the Column Definition screen:

COLUMN NAME	WIDTH	TYPE	UNIQUE KEY
NAME	30		X

---

COLUMN NAME	WIDTH	TYPE	UNIQUE KEY
STREET	30		
CITY	25		
STATE	2		
ZIP CODE	5		
PHONE NUMBER	13		X
POSITION APPLIED FOR	30		X
SALARY REQUIREMENT	7	C	
EVALUATION: STRONG, WEAK, MEDIUM	6		

---

Note that the unique key is made up of three columns. Specify the unique key by typing **x** in the Unique Key field for the NAME, PHONE NUMBER, and POSITION APPLIED FOR columns. Generate the APPLICANT table and load the rows shown above. You will use the JOB and APPLICANT tables that you have just created to define a view as described in the chapter "Deriving Views".





# Chapter 5: Deriving Views

---

## Derived from Stored Tables

Views are tables that are derived from one or more source tables or non-SQL defined records. When you create a view, you create a definition of how data already existing in the database is to be presented. When a view is accessed, the data is retrieved from the stored tables or non-SQL defined records in which it is stored and is presented in the format specified in the view definition. Updates to data made through a view are actually made to the source data that the view is presenting.

## Combine and Select as Needed

The relational operations (select, project, and join) can be used in any combination to create a view that meets your needs. You can derive a view by **joining** two or more source tables, **projecting** the columns from the source tables that you want to include in your view, and **selecting** rows from the source tables.

## Use Your Own or Others' Tables

ASF enables you to derive views easily, using existing tables as source tables. You can derive a view from tables that you own or from tables owned by others that have been made available to you through the COPY passkey (See the chapter "Granting Access to Tables").

Under the following topics, this section discusses how to derive a view from existing data tables:

- Defining views
- Using views
- View summary

This section contains the following topics:

[Defining Views](#) (see page 74)

[Using Views](#) (see page 86)

[View Summary](#) (see page 90)

## Defining Views

### The Steps

Defining a view involves some or all of the following steps:

1. Naming the view
2. Identifying the source table(s) and join criteria
3. Choosing columns
4. Defining additional columns
5. Selecting rows
6. Generating the view

### Example, Continued

To illustrate the steps involved in defining a view, the example presented in the chapter "Creating and Using Stored Tables" will be continued. In that chapter, you created the JOB and APPLICANT tables for the employment application that Fran Snell is developing for the personnel office of Acme Insurance. In this sample application, you will use those stored tables to derive the JOBS AND APPLICANTS view. This view will match applicants with open positions at Acme Insurance. To derive this view, you will perform the following relational operations:

- **Join** the JOB and APPLICANT tables based on position name.
- **Project** the desired columns.
- **Select** the positions that are open.

Follow steps 1 through 6 outlined below to derive the JOBS AND APPLICANTS view.

## Step 1: Naming the View

### Name the View

The Activity Selection screen is displayed. The JOB and APPLICANT tables should be listed in the table directory. Type the new table name, **jobs and applicants**, in the Table Name field:

```
TABLE NAME.: jobs and applicants
TABLE OWNER: SNELL
```

Remember that the table name can be any alphanumeric combination, including embedded spaces, of up to 32 characters.

Now that you have named the view, select the Define Table function ([PF2]) from the activity selection menu.

### Replace 'Stored' with 'View'

The Table Definition screen is now displayed. Notice that the table name JOBS AND APPLICANTS is in the Table Name field. Your user ID is in the Table Owner field. The View/Stored field below the Table Owner field contains the default value of STORED.

The cursor is positioned at the View/Stored field. Type **view** in place of STORED:

```

TDEF  CA - Automatic System Facility          CA          ** Table Definition **
DC560203 TABLE UNDEFINED

      _ PF1 - Help                          _ PF4 - Extended Table Definition
      _ PF2 - Define Columns                 _ PF5 - Delete Table Definition
      _ PF3 - Generate                       _ PF6 - Message Screen

Table Name.: JOBS AND APPLICANTS
Table Owner: SNELL                          Defn Number:
View/Stored: view                          Status.....: NEW
Comments...:

```

### ASF Adds a New Table

Press [Enter]. The message TABLE ADDED displays on the message display line. The Defn Number field to the right of the Table Owner field displays the definition number for the table. The Status field below the Defn Number field has changed from NEW to UNGENERATED.

Now you are ready to specify both the source tables from which the view is being derived and the criteria on which the source tables are being joined.

## Step 2: Specifying the Source Tables and Join Criteria

The Table Definition screen is still displayed. This screen enables you to perform a table derivation. Specify the following derivation information:

### Specify Source Table(s)

- Type **job** in the Table Name field under the Source Table #1 heading in the Table Derivation section of the screen. This indicates that the JOB table is your first source table.
- Type **applicant** in the Table Name field under the Source Table #2 heading. This indicates that the APPLICANT table is your second source table.

### Specify Join Columns

- Move the cursor to the Column #1 field. Enter the name of the column in the first source table that is related to a column in the second source table. This is the column on which you are *joining* the two tables. In this example, the TITLE column in the JOB table is related to the POSITION APPLIED FOR column in the APPLICANT table. Type **title** in the Column #1 field.
- Type **position applied for** in the Column #2 field.

Notice the statement Where Column #1 EQ Column #2 on the line with the Column #2 field. This indicates that you are joining the JOB table with the APPLICANT table, where the value of the TITLE column in JOB is the same as the value of the POSITION APPLIED FOR column in APPLICANT.

### Join Does not Require Equality

You may not always want to join tables on an equal condition. For example, you may want to join two tables when the value of one join column is greater than the value of the other join column. You can change the EQ operator to specify other join conditions, such as GT (greater than) or LT (less than).

Table Derivation	
Source Table #1	
Table Name.: <b>job1</b>	
Table Owner:	
Source Table #2	
Table Name.: <b>applicant2</b>	
Table Owner:	
Column #1: <b>title3</b>	
Column #2 <b>position applied for</b>	Where Column #1 EQ Column #2

1. Type JOB for the first source table.
2. Type APPLICANT for the second source table.
3. Enter TITLE and POSITION APPLIED FOR as the join columns.

- Accept the default join operator EQ.

### Select Define Columns

You have just defined the source tables and join criteria. You are now ready to define the columns for the JOBS AND APPLICANTS view. Select the Define Columns function ([PF2]) from the activity selection menu.

## Step 3: Choosing the Columns

The Column Definition screen is displayed, listing all the columns from the JOB and APPLICANT source tables. ASF automatically supplies these columns for the JOBS AND APPLICANTS view:

Column Name	Width	Type	Unique Key	Display Seq	Ext Screen
JOB NUMBER	4	TEXT		1	-
DEPARTMENT	20	TEXT		2	-
TITLE	30	TEXT		3	-
SALARY	7	CURRENCY		4	-
CURRENT STATUS (FILLED OR OPEN)	6	TEXT		5	-
NAME	30	TEXT		6	-
STREET	30	TEXT		7	-
CITY	25	TEXT		8	-
STATE	2	TEXT		9	-
ZIP CODE	5	TEXT		10	-
PHONE NUMBER	13	TEXT		11	-
POSITION APPLIED FOR	30	TEXT		12	-
SALARY REQUIREMENT	7	CURRENCY		13	-
EVALUATION: STRONG,WEAK,MEDIUM	6	TEXT		14	-

You can **project** a table that contains some or all of the source table columns by choosing the columns you want to include in the view. You can choose from the supplied column definitions in the following two ways:

- Delete columns you do not want to include in the table definition.
- Exclude columns that you want to include in the table definition, but that you do not want displayed on the Data Display screen.

### Delete Unused Columns

To continue this sample application, choose columns for the JOBS AND APPLICANTS view as follows:

- **Delete** the following columns from the table definition:
  - STREET
  - CITY
  - STATE
  - ZIP CODE
  - PHONE NUMBER

**Note:** Even though the phone number is a unique key in the APPLICANT source table, you can delete it from the view definition.

To delete each column:

- Move the cursor to the name of the column being deleted.
- Erase the column name.
- When you have erased the names of all the columns being deleted, press [Enter].

Erasing the column name causes the column definition to be deleted from the view (the column still exists in the source table). The Column Definition screen is rewritten to reflect the changes.

### Exclude Unviewed Columns

- **Exclude** the following columns from the screen display:
  - POSITION APPLIED FOR
  - CURRENT STATUS (FILLED OR OPEN)

To exclude each column from the screen display:

- Move the cursor to the Display Seq field for the column being excluded.
- Erase the display sequence number for the column.
- When you have erased the display sequence numbers of the columns you want to exclude from the table display, press [Enter].

Erasing the display sequence number of a column causes ASF to exclude the column from the screen display of the table; however, the columns remain part of the table definition.

**Exclude Columns Used, not Seen**

You are *excluding* these columns rather than *deleting* them because they are needed to define the view:

- The POSITION APPLIED FOR column is used to join the two source tables. Since it will always contain the same value as the TITLE column when the view is displayed, both columns do not need to be displayed.
- The CURRENT STATUS column will be used to select rows for the view as part of the definition. You are excluding the CURRENT STATUS column because, once the selection condition is defined, it will always contain the value OPEN.

**Renumber Columns**

When you have deleted and excluded the appropriate columns, renumber the display sequence of the remaining columns. The process of projecting columns for the JOBS AND APPLICANTS view is illustrated below.

Column Name	Width	Type	Unique Key	Display Seq	Ext Screen
JOB NUMBER	4	TEXT		1	-
DEPARTMENT	20	TEXT		2	-
TITLE	30	TEXT		3	-
SALARY	7	CURRENCY		4	-
CURRENT STATUS (FILLED OR OPEN)	6	TEXT		5	-
NAME	30	TEXT		6	-
	30	TEXT		7	-
	25	TEXT		8	-
	2	TEXT		9	-
	5	TEXT		10	-
POSITION APPLIED FOR	13	TEXT	11	-	-
SALARY REQUIREMENT	30	TEXT		12	-
EVALUATION: STRONG,WEAK,MEDIUM	7	CURRENCY		13	-
	6	TEXT		14	-

Column Name	Width	Type	Unique Key	Display Seq	Ext Screen
JOB NUMBER	4	TEXT		2	-
DEPARTMENT	20	TEXT		4	-
TITLE	30	TEXT		6	-
SALARY	7	CURRENCY		8	-
CURRENT STATUS (FILLED OR OPEN)	6	TEXT			-
NAME	30	TEXT		10	-
POSITION APPLIED FOR	30	TEXT			-
SALARY REQUIREMENT	7	CURRENCY		12	-
EVALUATION: STRONG,WEAK,MEDIUM	6	TEXT		14	-



```

CA
CDEF CA - Automatic System Facility      ** Column Definition **
DC560302 ADD/MODIFY COLUMNS
Table Name: JOBS AND APPLICANTS

  _ PF1 - Help                          _ PF3 - Extended Selection Definition
  _ PF2 - Extended Column Definition     _ PF4 - Extended Key Definition

Page: 1 of 1

```

Column Name	Width	Type	Unique Key	Display Seq	Ext Screen
JOB NUMBER	4	TEXT		2	-
DEPARTMENT	20	TEXT		4	-
TITLE	30	TEXT		6	-
SALARY	7	CURRENCY		8	-
CURRENT STATUS (FILLED OR OPEN)	6	TEXT			-
NAME	30	TEXT		10	-
POSITION APPLIED FOR	30	TEXT			-
SALARY REQUIREMENT	7	CURRENCY		12	-
EVALUATION: STRONG,WEAK,MEDIUM	6	TEXT		14	-

## Step 4: Adding a Column

### Added Columns not Stored

You might want to add a column to a view so you can perform calculations on data supplied by the view. The added column is for display purposes only; it is not stored in the database because it is not part of the source tables from which the view is derived.

### Define the Column

To continue the example, add the SALARY DIFFERENTIAL column to the JOBS AND APPLICANTS view, as follows:

- Type **salary differential** in the next available Column Name field (after the EVALUATION: STRONG, WEAK, MEDIUM column).
- Enter a WIDTH of **7** for the SALARY DIFFERENTIAL column.
- Enter a TYPE of **c** (for CURRENCY).
- Enter a display sequence number of **16** for the column in the Display Seq field.
- Tab to the Ext Screen field for the SALARY DIFFERENTIAL column and type **x**.
- Select the Extended Column Definition function ([PF2]).

Column Name	Width	Type	Unique Key	Display Seq	Ext Screen
JOB NUMBER	4	TEXT		2	-
DEPARTMENT	20	TEXT		4	-
TITLE	30	TEXT		6	-
SALARY	7	CURRENCY		8	-
CURRENT STATUS (FILLED OR OPEN)	6	TEXT			-
NAME	30	TEXT		10	-
POSITION APPLIED FOR	30	TEXT			-
SALARY REQUIREMENT	7	CURRENCY		12	-
EVALUATION: STRONG,WEAK,MEDIUM	6	TEXT		14	-
salary differential	7	c		16	x

**Extended Column Definition**

The Extended Column Definition (XCOL) screen is displayed. This screen shows the column characteristics that are supplied by ASF when you define a column. For most columns, you do not need to review the extended definition; the characteristics supplied by ASF are sufficient for the column definition. For columns added to views, you have to supply additional information to fully define the column.

**Define the Calculation**

You want the SALARY DIFFERENTIAL column to calculate the difference between the salary an applicant requires and the salary available for the position. Define the SALARY DIFFERENTIAL column by entering the arithmetic expression **column(salary requirement) - salary** in the Calculation field.

**Embedded Spaces**

Since the SALARY REQUIREMENT column name contains an embedded space, you must refer to it using COLUMN(column-name).

For more information on specifying calculations, see the chapter "Extended Functions of ASF".

```

XCOL  CA - Automatic System Facility          CA
DC560407 MODIFY COLUMN DEFINITION          *Extended Column Definition*
Table Name: JOBS AND APPLICANTS
      _ PF1 - Help      _ PF2 - Next Column      _ PF3 - Message Screen

      Displayed Column Name          Stored Column Name
      SALARY DIFFERENTIAL          SALARY-DIFFERENTIAL

      Display Format          Stored Format
      -Z(05).9(02)          S9(05)V9(02)

Usage: 3 D/0/1/2/3 Display/Comp/Comp-1/Comp-2/Comp-3      IDD Version:
Comments.....:          Column Level: WORK
Initial Value:
Calculation..: column(salary requirement) - salary

Edit Table:          Version:          Code Table:          Version:
  U  U/P  Unprotected/Protected  N  B/R/P/T/G/Y/W/N  Blue/Red/Pink/Turquoise
  r  Y/N  Blank When Zero          N  N/R  Normal/Reverse Video
  D  B/D/K  Bright/Display/Dark  N  Y/N  Blink/NoBlink

```

### What You Will See

Whenever the JOBS AND APPLICANTS view is accessed, ASF will subtract the value in the SALARY column from the value in the SALARY REQUIREMENT column and display the result in the SALARY DIFFERENTIAL column.

Press [Enter]. The message COLUMN DEFINITION MODIFIED is displayed on the message display line. Press [PF2] to return to the Column Definition screen.

## Step 5: Selecting Rows

As part of the view definition, you can specify criteria to **select** rows from the source tables. Only the rows that meet the criteria specified will be included in the view.

### Open Jobs Only

In the JOBS AND APPLICANTS view, you want to include only the positions that are open, so you want to select the rows in which the CURRENT STATUS (FILLED OR OPEN) column has a value of OPEN. To specify criteria for selecting these rows, select the Extended Selection Definition function ([PF3]) from the Column Definition screen.

### Extended Selection Definition

The Extended Selection Definition (XSEL) screen is displayed. This screen looks exactly like the Data Selection screen you used in . Use of the two screens is exactly the same as well. The selection criteria you enter on this screen, however, become part of the table definition.

The column names of the view are listed below the Column Name field heading. Each column has been assigned an identification number, which is listed below the ID field heading to the left of the Column Name field. You can use the ID numbers when specifying selection criteria.

### Define Selection Criteria

Define the selection criteria for the JOBS AND APPLICANTS view, as follows:

- Move the cursor to the Selection Criteria field next to the CURRENT STATUS (FILLED OR OPEN) column name.
- Type 'open' in single quotation marks.

### EQ is the Default Selector

You do not need to include the conditional operator EQ in the selection criteria because it is assumed by ASF. You can also select rows on other conditions, such as less than (LT), greater than (GT), or not equal (NE). When using other conditions, you must specify the conditional operator. For example, you can enter **ne 'open'** to select the rows where the CURRENT STATUS column entry is not equal to OPEN.

```

                                     CA
XSEL  CA - Automatic System Facility  *Extended Selection Definition*
DC560603 ENTER SELECTION CRITERIA
Table Name: JOBS AND APPLICANTS
          _ PF1 - Help      _ PF2 - Message Screen      Page: 1 of 1

  ID      Column Name                Selection Criteria
0001 JOB NUMBER
0002 DEPARTMENT
0003 TITLE
0004 SALARY
0005 CURRENT STATUS (FILLED OR OPEN) 'open'
0006 NAME
0007 PHONE NUMBER
0008 POSITION APPLIED FOR
0009 SALARY REQUIREMENT
0010 EVALUATION: STRONG,WEAK,MEDIUM
0011 SALARY DIFFERENTIAL

                                     Additional Selection Criteria
```

**Enter Criteria**

Press [Enter]. The message SELECTION UPDATED is displayed, indicating that the selection criteria have been accepted by ASF.

**Note:** If you make an error in entering the selection criteria, the erroneous items are highlighted and you are instructed to access the Message screen ([PF2]). The Message screen lists the syntax errors that need to be corrected.

Press [Clear] to return to the Column Definition screen.

## Step 6: Generating the View

When you generate a view through ASF, the view definition is completed and an application is created for accessing the view. When the view is accessed, the data is retrieved from the source tables from which the view is derived.

**Use Table Definition Screen**

To generate the JOBS AND APPLICANTS view, press [Clear] from the Column Definition screen to access the Table Definition screen. Select the Generate function ([PF3]) from the activity selection menu on the Table Definition screen.

The Table Definition screen remains displayed while ASF generates the view. When the generation is complete, the Status field changes from UNGENERATED to GENERATED, and the message display field displays the message TABLE GENERATED SUCCESSFULLY.

**Note:** If the generation is unsuccessful, you are instructed to access the Message screen ([PF6]). The Message screen lists all the errors that caused the table generation to fail.

Now that the view is generated, you can use the view through the online application that ASF generated for it.

## Using Views

The application created when you generate a view enables you to:

- Display rows
- Select rows
- Change rows

You create a view to access data stored in the database and display that data in a new way. The data presented in a view is accessed directly from the source tables. When you change data presented in a view, you are actually modifying the data stored in the source tables.

Follow steps 1 through 3 to use the JOBS AND APPLICANTS view.

### Step 1: Displaying Rows

#### Use Display/Change Data screen

To display the JOBS AND APPLICANTS view you have just generated, return to the Activity Selection screen ([PA1]) and select the Display/Change Data function.

PF1 = ADD; PF2 = CHANGE; PF3 = DELETE; ENTER = NEXT; CLEAR = EXIT	
JOBS AND APPLICANTS	
JOB NUMBER	C417
DEPARTMENT	CLAIMS
TITLE	CLAIMS ADJUSTER
SALARY	<b>17000.00</b>
NAME	JOHN BURCHILL
SALARY REQUIREMENT	<b>20000.00</b>
EVALUATION: STRONG,WEAK,MEDIUM	WEAK
SALARY DIFFERENTIAL	<b>3000.00</b>

Notice that the POSITION APPLIED FOR and CURRENT STATUS (FILLED OR OPEN) columns do not appear in the screen display. Note also that the SALARY DIFFERENTIAL column contains the difference between the SALARY REQUIREMENT and SALARY column entries.

Press [Enter] to display the next row. Continue to press [Enter] to look at all of the rows in the view. Press [Clear] to return to the Activity Selection screen.

---

## Step 2: Selecting Rows

### Use Data Selection Screen

The Activity Selection screen is displayed. Select the Select Data function ([PF5]) from the activity selection menu.

The Data Selection (DSEL) screen is displayed. This screen enables you to specify selection criteria for displaying rows of the view. ASF will display only the rows that match the criteria you specify.

The column names of the view are listed below the Column Name field heading. Each column has been assigned an identification number, which is listed below the ID field heading to the left of the Column Name field. You can use the ID numbers in place of column names when specifying selection criteria.

### Required Salary &le. Salary

In this example, you want to select the rows from the JOBS AND APPLICANTS view in which the salary required by the applicant is less than or equal to the salary being offered for the position.

The cursor is positioned at the first field under the Selection Criteria field heading. Press [Return] to move the cursor to the SALARY REQUIREMENT column name. Type the selection criterion **le #4** in this field. Note that the ID number 4, preceded by a pound sign (#), enables you to reference the SALARY column name in your selection criterion.

```

DSEL  CA - Automatic System Facility          CA
DC560603 ENTER SELECTION CRITERIA          ** Data Selection **
Table Name: JOBS AND APPLICANTS
      _ PF1 - Help      _ PF2 - Message Screen      Page: 1 of 1

  ID      Column Name      Selection Criteria
0001 JOB NUMBER
0002 DEPARTMENT
0003 TITLE
0004 SALARY
0005 CURRENT STATUS (FILLED OR OPEN)
0006 NAME
0007 PHONE NUMBER
0008 POSITION APPLIED FOR
0009 SALARY REQUIREMENT      le #4
0010 EVALUATION: STRONG,WEAK,MEDIUM
0011 SALARY DIFFERENTIAL

Additional Selection Criteria

```

You have just specified that you want to display the rows of the JOBS AND APPLICANTS view in which the entry in the SALARY REQUIREMENT column is less than or equal to the entry in the SALARY column.

#### ASF Uses your Selection

Press [Enter]. ASF saves the selection criteria you have entered, and the message SELECTION MODIFIED, HIT ENTER TO PROCEED TO DATA SELECTION is displayed. Press [Enter] again.

The Data Display screen is displayed automatically. It displays the columns of the JOBS AND APPLICANTS table and the first row that matches the selection criteria you specified. Notice that the value in the SALARY REQUIREMENT column is less than or equal to the value in the SALARY column and that the SALARY DIFFERENTIAL column contains a value that is less than or equal to zero.

**Note:** If you make an error when entering the selection criteria, you remain on the Data Selection screen. The field names to which the erroneous selection criteria apply are highlighted. The message display line instructs you to access the Message screen, which lists all the error conditions.



CA	
PF1 = ADD; PF2 = CHANGE; PF3 = DELETE; ENTER = NEXT; CLEAR = EXIT	
JOBS AND APPLICANTS	
JOB NUMBER	0198
DEPARTMENT	OFFICE SERVICES
TITLE	MAIL CLERK
SALARY	7000.00
NAME	TIMOTHY MURRAY
SALARY REQUIREMENT	6000.00
EVALUATION: STRONG,WEAK,MEDIUM	STRONG
SALARY DIFFERENTIAL	-1000.00

Press [Enter] to display the next selected row. Continue to press [Enter] until all the rows you selected have been displayed.

Press [Enter] again to redisplay the first selected row.

## Step 3: Changing Rows

### Changes Made to Stored Table(s)

With the online application created for a view you can display the rows of the view and make changes to them. The changes are made directly to the stored tables from which the view is derived.

### You Cannot Add or Delete Rows

You cannot add rows to a view or delete rows from a view; additions and deletions can be made only to stored tables.

For this sample application, change the value in the SALARY column. Press [PF2] to modify the source table in the database. Notice that the value of the SALARY DIFFERENTIAL column is recalculated when the row is modified.

### You Cannot Make Join Criteria Untrue

When modifying data through a view, keep in mind that you cannot change an entry in a join column so that the join condition is no longer true. For example, in this sample application you cannot change the TITLE column because it must equal the POSITION APPLIED FOR column, which is not displayed.

### ASF Processes Changes

Return to the Activity Selection screen ([PA1]) and access the JOB table (type **job** in the Table Name field and press [Enter]). Display the job that you just modified through the JOBS AND APPLICANTS view. Notice that the SALARY column reflects the change that you made through the view.

## View Summary

The following list summarizes the steps involved in deriving a view through the Automatic System Facility:

1. **Name the view** by typing a unique name in the Table Name field on the Activity Selection screen.
2. **Designate the table as a VIEW** by selecting the Define Table function ([PF2]) from the Activity Selection screen and entering **view** in the View/Stored field on the Table Definition screen.
3. **Specify the source tables and the join criteria** by typing the name of the first source table in the Table Name field under the Source Table #1 heading, and the name of the second source table in the Table Name field under the Source Table #2 heading. Type the names of the columns on which the two tables are to be joined.
4. **Choose the columns** by deleting columns from the table definition or by excluding columns from the screen display. **Delete** columns by erasing the column names; **exclude** columns by erasing the columns' display sequence numbers.
5. **Add columns** by entering column names and widths and, optionally, column entry types on the Column Definition screen. Access the Extended Column Definition screen for the columns added and specify a CALCULATION that defines each column. Columns added to views are work columns. The data displayed in a work column is not stored in the database.
6. **Select the rows** to be included in the view by selecting the Extended Selection Definition function ([PF3]) from the Column Definition screen. Define selection conditions on the Extended Selection Definition screen.
7. **Generate the view** by returning to the Table Definition screen and selecting the Generate function ([PF3]).
8. **Display, select, and change rows** in the view by selecting the Display/Change Data ([PF4]) and Select Data ([PF5]) functions from the Activity Selection screen.

# Chapter 6: Granting Access to Tables

---

## Passkeys Give Access

You can allow other users to access tables that you create by assigning **passkeys**. A passkey defines the type of access allowed. Two types of access are granted by passkeys:

- **Data access** allows access to tables (data and definitions). For example, the ADD data access passkey allows a user to add rows to a table; the REDEFINE data access passkey allows a user to modify the definition of a table.
- **Catalog access** allows access to the catalog, which is the directory of users and tables. For example, the BROWSE catalog access passkey allows a user to list the directory of another user or, in conjunction with data access passkeys, to access an individual table in another user's directory.

ASF provides individual passkeys that allow you to be very specific in assigning data and catalog access. Additionally, ASF provides **predefined passkey combinations** that enable you to specify useful combinations of data and catalog access with a single passkey.

For more information about each passkey type, see the chapter "Passkey Screens".

## Row-level Security

In addition to passkeys that restrict or allow access to entire tables, you can use row-level security to restrict access to separate rows. Row-level security is discussed in the chapter "Restricting Access to Rows".

This section discusses how to grant other users access to your tables and covers the following topics:

- Assigning passkeys
- Passkey summary

This section contains the following topics:

- [Assigning Passkeys](#) (see page 92)
- [Passkey Summary](#) (see page 100)

## Assigning Passkeys

### Example Continued

To illustrate how you can assign passkeys through ASF, the example presented in the chapter "Creating and Storing Used Tables" and "Deriving Views", will be continued. In this section, you will follow Fran Snell as she performs the following passkey activities:

- Giving passkeys
- Displaying passkeys
- Deleting passkeys

## Giving Passkeys

### Different Access for Different People

Now that she has developed the employment application, which consists of the JOB and APPLICANT stored tables and the JOBS AND APPLICANTS view, Fran wants to delegate the responsibility of keeping the data updated. She wants to assign passkeys to other ASF users in the personnel department so they can use and update the employment application. Fran selects the following individuals to use her employment application:

- Karen Cross, who updates job information at Acme Insurance
- John Forrest, who processes applications for positions at Acme Insurance
- Stan White, who schedules interviews for employment candidates
- Allen Fites, who, as Fran's supervisor, is given access to all of Fran's tables

Fran takes the following steps to assign passkeys to the users listed above:

1. Selects the Passkey and Give Passkeys functions
2. Gives data access and BROWSE passkeys
3. Gives a predefined passkey combination

## Step 1: Selecting the Passkey and Give Passkeys Functions

The Activity Selection screen is displayed. Fran selects the Passkey function from the activity selection menu by pressing [PF9]. The Passkey Function screen, which provides a menu of passkey activities, is displayed.

From the menu listed on the Passkey Function screen, Fran chooses the Give Passkeys function by typing x next to that listed selection.

```

PASS  CA - Automatic System Facility      CA      *** Passkey Function ***
DC422007 SELECT AN ACTION

                                     _ PF1 - Help

Select Action To Be Performed:

  x Give Passkeys
  _ Delete Passkeys
  _ Show Passkeys Given
  _ Show Passkeys Held

```

## Step 2: Giving Data Access and BROWSE Passkeys

The Give Passkeys screen lists all the passkey types in three columns: the data access passkeys are on the left, the catalog access passkeys are in the middle, and the predefined passkey combinations are listed on the right.

### Determine Needs

Passkeys can be assigned either for individual tables, a user's ID (gives access to all of the user's tables), or a folder (CA-ICMS users only). Fran wants to assign data access passkeys for individual tables. Fran determines each user's needs as follows:

- Karen Cross needs to add and change rows in the JOB table.
- John Forrest needs to add and change rows in the APPLICANT table.
- Stan White needs to display the JOBS AND APPLICANTS view and to derive the INTERVIEW SCHEDULE view from the APPLICANT table.

The **data access passkeys** that Fran assigns to each user are shown below:

Give the passkeys to this user	For access to this table	Passkey types
CROSS	JOB	ADD MODIFY
FORREST	APPLICANT	ADD MODIFY

Give the passkeys to this user	For access to this table	Passkey types
WHITE	JOB AND APPLICANTS	LIST
	APPLICANT	COPY

### Need BROWSE for Data Access

In addition to the data access passkeys, Fran must also give the users a **BROWSE passkey** that enables them to access the individual tables from her directory. The BROWSE passkey is a catalog access passkey that allows users to select tables from a directory. You must give a user a BROWSE passkey, which allows access to your directory, before the user can access any tables in your directory. Like the data access passkeys, the BROWSE passkey can be given for an individual table or for your user name.

When given for an individual table, the BROWSE passkey must be accompanied by data access passkeys in order to access the table. When given for a user name, the BROWSE passkey enables you to list the contents of the user's directory. Again, in order to access any of the tables in the directory, you must have data access passkeys for the tables.

In this example, Fran decides to give data access and BROWSE passkeys for individual tables. She begins by assigning passkeys to user CROSS for access to the JOB table:

1. Fran types **x** next to Add, Modify, and Browse to give those passkeys.
2. She moves the cursor to the Give The Passkey(s) To This User Or Group field and types the name of the user to whom she is assigning passkeys; Fran types **cross** in this field.
3. In the For Access To The Following Entry field, Fran types the table name **job**.
4. She leaves the default value **snell** in the The Owner Of The Entry Is field because user SNELL is the owner of the JOB table.

```

GPAS  CA - Automatic System Facility          CA
DC422007 SELECT AN ACTION                    *** Give Passkeys ***

                                           _ PF1 - Help

Select Passkey Type(s):

_ Copy          x Browse          _ DBA
_ Create        _ Management      _ Read
x Add           _                 _ Alter
x Modify        _                 _ Write
_ Erase         _                 _ Update
_ Redefine     _                 _ All
_ List

Give The Passkey(s) To This User Or Group:

cross

For Access To The Following Entry:

job

The Owner Of The Entry Is: SNELL

```

Fran presses [Enter] to give the ADD, MODIFY, and BROWSE passkeys to ASF user Karen Cross. Karen Cross now has access to Fran's JOB table. She can display the table definition and the table's rows, and she can add and modify rows in the table.

To give the passkeys to the other ASF users, Fran changes the appropriate entries on the Give Passkeys screen to give John Forrest access to the APPLICANT table and give Stan White access to the APPLICANT table and to the JOBS AND APPLICANTS view.

### Step 3: Giving a Predefined Passkey Combination

Fran wants to grant Allen Fites access to all the tables she owns. She wants him to have access to her directory through the BROWSE passkey and to have complete data access to all of her tables. She can grant both data and catalog access by assigning one predefined passkey combination.

Fran decides to give user FITES a WRITE passkey. Fran types **x** next to the WRITE passkey.

Fran types the user name **fites** in the Give The Passkey(s) To This User Or Group field. She enters her user name in the For Access To The Following Entry field.

```
GPAS  CA - Automatic System Facility      CA      *** Give Passkeys ***
DC422018 ENTITY NOT FOUND: CROSS

                                     _ PF1 - Help

Select Passkey Type(s):

_ Copy          _ Browse          _ DBA
_ Create        _ Management      _ Read
_ Add           _                 _ Alter
_ Modify        _                 x Write
_ Erase         _                 _ Update
_ Redefine     _                 _ All
_ List

Give The Passkey(s) To This User Or Group:

fites

For Access To The Following Entry:

snell

The Owner Of The Entry Is: SNELL
```

Since Fran gave the passkey for access to her user name, the passkey applies to all of the tables Fran owns. With the WRITE passkey Fran has given him, Allen Fites can perform the following activities:

- Browse through Fran's directory
- Copy Fran's tables for derivation
- Create tables for Fran
- Add rows to her tables
- Modify rows in her tables
- Erase her tables
- Redefine her tables



## Displaying Passkeys

Fran now wants to review the passkey assignments she has made and display the passkey assignments others have made for her. To display passkey assignments, Fran takes the following steps:

1. Shows passkeys she gave to various users to access her tables
2. Shows passkeys she holds to access other users' tables

Each of these steps is discussed separately on the following pages.

### Step 1: Showing Passkeys Given

Fran presses [Clear] to return to the Passkey Function screen from the Give Passkeys screen. Fran selects the Show Passkeys Given function from the menu on the Passkey Function screen.

The Show Passkeys Given screen is displayed. This screen enables Fran to specify a user name or table name, and requests ASF to display the users who have access to the user or table specified.

Fran enters the table name APPLICANT in the Entry Name field. Since she owns the table, she leaves the default value SNELL in the Owner Name field.

When Fran presses [Enter], the Passkeys Given screen is displayed. This screen lists the users who have passkeys for the APPLICANT table, and shows which passkeys each user holds.

```
PASG CA - Automatic System Facility CA *** Show Passkeys Given ***
DC422007 SELECT AN ACTION
_ PF1 - Help

Entry Name: applicant
Owner Name: SNELL

Requested Action:

Show The Users And Groups Which Have Access To The Entry Which Is
Specified Above.
```

```

PASH CA - Automatic System Facility CA *** Passkeys Given ***

List Of Passkeys Given For:
APPLICANT

Entry Name          Entry Type          Passkeys
                    USER              B L C C A M E R M
                    USER              R S P R D O R D A
                    USER              S T Y E D D S F N

WHITE              USER              Y  Y  Y
FORREST           USER              Y   Y Y
    
```

Fran presses [Clear] to return to the Show Passkeys Given screen. She returns to the Passkey Function screen by entering PASS in the screen name field and pressing [Enter].

### Step 2: Showing Passkeys Held

The Passkey Function screen is displayed. Fran selects the Show Passkeys Held function from the menu.

The Show Passkeys Held screen is displayed. This screen enables Fran to request a list of the passkeys she holds for tables owned by other users.

To request that ASF show the users and tables to which she has access, Fran leaves the default value SNELL in the Entry Name field and presses [Enter].

When Fran presses [Enter], the Passkeys Held screen is displayed. This screen shows the tables and users to which Fran has access, and the type of access she has.

```

PASH CA - Automatic System Facility CA *** Show Passkeys Held ***
DC422007 SELECT AN ACTION

_ PF1 - Help

Entry Name: SNELL

Requested Action:

Show The Entries To Which The User Or Group Specified Above Has Access.
    
```

PASH CA - Automatic System Facility		CA	*** Passkeys Held ***							
List Of Passkeys Held By:			SNELL							
Entry Name	Entry Type	Passkeys								
		B	L	C	C	A	M	E	R	M
		R	S	P	R	D	O	R	D	A
		S	T	Y	E	D	D	S	F	N
CROSS										
Owner: CORP	USER	Y	Y	Y	Y	Y	Y	Y	Y	Y
FITES										
Owner: CORP	USER	Y								
FORREST										
Owner: CORP	USER	Y	Y	Y	Y	Y	Y	Y	Y	Y
MONTHLY REPORT										
Owner: FITES	DATA TABLE							Y	Y	Y
WHITE										
Owner: CORP	USER	Y	Y	Y	Y	Y	Y	Y	Y	Y

## Deleting Passkeys

As needs change, Fran can adjust the passkey assignments she has made. For example, when Karen Cross moves to another department in Acme, Fran wants to remove the passkeys she has given Karen. To remove passkeys that have been given, Fran selects the Delete Passkeys function from the menu on the Passkey Function screen.

The Delete Passkeys screen is displayed. This screen enables Fran to delete the passkeys she has given to Karen Cross. Fran gave Karen passkeys that allowed her to access the JOB table. Since Fran wants to delete all the passkeys she has given to Karen, she completes the Delete Passkeys screen as follows:

- She types X next to each of the passkeys she had given to Karen:
  - Add
  - Modify
  - Browse
- She types CROSS in the Delete The Passkey(s) From This User Or Group field.
- She types JOB in the For Access To The Following Entry field.
- She leaves the default value SNELL in the The Owner Of The Entry Is field.

**Note:** When deleting passkeys, you must specify only passkeys that have been assigned previously, or no passkeys are deleted. For example, unless you have assigned all passkeys to a user, you cannot specify ALL when deleting passkeys.

```
DPAS CA - Automatic System Facility          CA
DC422007 SELECT AN ACTION                    *** Delete Passkeys ***

                                          _ PF1 - Help

Select Passkey Type(s):

_ Copy          x Browse          _ DBA
_ Create        _ Management      _ Read
x Add           _                 _ Alter
x Modify        _                 _ Write
_ Erase         _                 _ Upate
_ Redefine     x All
_ List

Delete The Passkey(s) From This User Or Group:

cross

For Access To The Following Entry:

job

The Owner Of The Entry Is: SNELL
```

## Passkey Summary

Passkeys offer ASF users control over the data they store in the database. With passkey security, you can decide who can access your tables and what type of access they are allowed. Additionally, the ASF administrator can place global passkey restrictions on users, thereby restricting the activities the users can perform through ASF.

You have full passkey authority, barring any global restrictions, over tables you own, and you can assign passkeys to other users that allow them to access those tables. The passkey functions of ASF enable you to perform the following activities:

- Give passkeys to other users
- Delete passkeys you have given to other users
- Show passkeys you have given to other users
- Show passkeys other users have given to you

When CA ICMS is installed, users achieve the same results when they perform passkey functions through ASF as when they perform passkey functions through CA-ICMS. For example, passkeys can be assigned to groups as well as to users.

For more information about each passkey type, see the chapter "Passkey Screens".

# Chapter 7: Restricting Access to Rows

---

Just as access to an entire table or view is controlled by the assignment of passkeys, access to **rows** within a table is controlled by **row-level security**. Access to rows within a data table is controlled by the association between security names, tables, and users.

## Security Names Control Row Access

Row-level security involves assigning a **security name** to one or more tables and associating that security name with a user. By matching security names of tables with users, the system can tell which users are allowed to access which tables. You can also associate selection criteria with the security name. The selection criteria tells ASF which rows of the table the associated user is allowed to see.

## DBA allows Row-level Security

**To implement row-level security**, users must be authorized to use row-level security. Authorization is granted by the database administrator (DBA). The DBA is authorized to access row-level security upon installation of ASF. Generally, the DBA will maintain row-level security on corporate objects while users maintain the security on privately owned objects. For more information on initializing row-level security in the ASF environment and authorizing users to access row-level security, see the chapter "Administrative functions of ASF".

This section contains the following topics:

[Assigning Row-level Security to Data Tables](#) (see page 102)

[Row-level Security Summary](#) (see page 112)

## Assigning Row-level Security to Data Tables

Row-level security is provided through system tables that keep track of the relationship between each security name, table, and user.

### Three Tables Maintain Security

- The **\$SECURITY-RUNTIME-TABLES\$** (SRT) is a table through which you associate the security name with users and WHERE criteria. The figure below shows several rows that have been entered in an SRT.

GROUP	USER	OWNER	SECURITY-NAME	STATUS	WHERE-CLAUSE
DBA GROUP	MON	IIZ	SECNAME	(V)	DEPTID EQ 'SLS'
SYS GROUP	BRM	LPN	DOC	(V)	TECH EQ 'PUBS'
SYSTECH	DPW	JJM	AUD	(V)	TAX EQ 'IRS'
TECHREV	LLM	BDR	RULE	(V)	ACT EQ 'AUD'

- The **\$OBJECT-SECURITY-NAME-TABLES\$** (OST) is a table through which the table name, owner name, security name, and table-definition number of the table to which you are applying row-level security are registered. This information is maintained internally by ASF.
- The **\$SRT-OST-CROSS-REFERENCES\$** (SXRF) combines information from the SRT and the OST and allows you to review the security names and criteria you have associated with your own tables.

When you assign row-level security to a table, the information is stored in these tables. When a user tries to access a table protected by row-level security, ASF checks these tables to see if that user is allowed to access the table.

### Assigning Security

The steps to assigning row-level security are:

- **Assign a security name** to a table. Security names tell the system that row-level security is in effect.
- **Associate users and WHERE criteria with a security name.** This limits access to data in tables associated with a specific security name.

With row-level security, you can also:

- **Assign security names at the group level**, to associate security names and WHERE criteria with more than one user.
- **Assign generic names** to columns that contain the same type of data in different tables. This enables you to create a single entry in the \$SECURITY-RUNTIME-TABLE\$ with the same restrictions that apply to two or more tables.

#### **Regenerate the Table(s)**

Assigning row-level security to a table changes the definition of that table. In order to access the stored data after applying row-level security, you must regenerate the table. You should, however, wait until you are finished assigning row-level security to the table; including assigning WHERE criteria and generic column names as described later in this section.

**Note:** This regeneration does not cause data conversion.

## **Assigning Row-level Security**

Follow the steps presented below to restrict access to rows of a stored table. Steps 1 and 2 indicate how to assign row-level security for a single user. Steps 3 and 4 indicate how to assign row-level security for multiple users.

### **Step 1: Assigning a Security Name to a Table**

Fran Snell's APPLICANT table contains information about job applicants and their salary requirements. Fran has already assigned passkeys to Forrest that allow him to modify and add to this table. To be sure that the salary information over \$50,000 is not accessible to Forrest, Fran puts row-level security on the table.

To assign a security name to a table, perform the following steps:

1. Enter APPLICANT in the Table Name field of the Activity Selection (ASEL) screen.
2. Type RSEC in the screen name field of the ASEL screen to access the Row-Level Security screen:

```

rsec CA - Automatic System Facility CA ** Activity Selection **
DC560004 SELECT A TABLE

User Name: SNELL

_ PF1 - Help          _ PF5 - Select Data    _ PF13 - Query
_ PF2 - Define Table  _ PF7 - Page Backward _ PF14 - Signon
_ PF3 - Load Data    _ PF8 - Page Forward  _ PA1  - Prior Level
_ PF4 - Display/Change Data _ PF9 - Passkey      _ CLEAR - Leave ASF
_ PF15 - Administrator

Table Name.: applicant
Table Owner: SNELL

Page: 1 of 1

_ APPLICANT
_ JOB
_ JOB AND APPLICANT
    
```

3. Press [Enter] to access the RSEC screen. Once on the RSEC screen, specify the security name of the table (SALSEC) next to the Security Name field:

```

RSEC CA - Automatic System Facility CA ** Row Level Security **
DC560314 ADD/MODIFY SECURITY NAME AND GENERIC COLUMN NAMES
Table Name: APPLICANT

_ PF1 - Help Page: 1 of 1

Security Name: salsec

Column Name          Generic Column Name
NAME
STREET
CITY
STATE
ZIP CODE
PHONE NUMBER
POSITION APPLIED FOR
SALARY REQUIREMENT
EVALUATION: STRONG, WEAK, MEDIUM
    
```

4. Press [Enter] to allow ASF to record the security name in the SRT table.



## Step 2: Associating a User and WHERE Criteria with the Security Name

To restrict access to the table by Forrest and prevent him from viewing salary information exceeding \$50,000, Fran Snell wants to specify WHERE criteria. If no criteria are specified, Forrest will be able to view the entire APPLICANT table including all the salaries.

To enter WHERE criteria for the table, perform the following steps:

1. Type RLOD in the screen name field of any screen and press [Enter] to access the Load Data option for the SRT:

	CA
PF1 = ADD;PF2 = CHANGE;PF3 = DELETE;PF4 = VALIDATE;ENTER = NEXT;CLEAR = EXIT	
\$SECURITY-RUNTIME-TABLE\$	
GROUP	
USER	
OWNER	
SECURITY-NAME	
STATUS	
WHERE-CLAUSE	

Specify the following information on the RLOD screen:

- **USER**—The user ID of Forrest (FORREST) so he can access the table.
- **OWNER**—Snell's user ID (SNELL).
- **SECURITY-NAME**—The security name of the table to which you are applying row-level security (SALSEC).
- **WHERE-CLAUSE**—The restrictions that define the data Forrest will be allowed to see (COL(SALARY REQUIREMENT) LT 50000).

	CA
PF1 = ADD;PF2 = CHANGE;PF3 = DELETE;PF4 = VALIDATE;ENTER = NEXT;CLEAR = EXIT	
\$SECURITY-RUNTIME-TABLE\$	
GROUP	
USER	forrest
OWNER	snell
SECURITY-NAME	salsec
STATUS	
WHERE-CLAUSE	col(salary requirement) lt 50000

2. Press [PF1] to add the row.
3. Press [PF4] to validate the entry:

NO ERRORS ON VALIDATION	CA
\$SECURITY-RUNTIME-TABLE\$	
GROUP	
USER	
OWNER	
SECURITY-NAME	
STATUS	
WHERE-CLAUSE	

## WHERE Criteria Considerations

The **WHERE** criteria specified must conform to the format of the selection criteria defined on the Data Selection screen. When specifying WHERE criteria, the following considerations apply:

- Omitting **WHERE** criteria permits access to the entire table.
- You can specify **WHERE** criteria for a view. When the view is accessed, the criteria specified for its source tables and its own criteria are combined. For example, suppose one source table has criteria of A LT B, the other source table has criteria of B GT C, and the view has criteria of F NE 10. At run time, the criteria for the view will be processed as: A LT B *and* B GT C *and* F NE 10.
- Columns that contain special characters (including hyphens, spaces, and periods) must be enclosed in parentheses following the literal column, which can be abbreviated to **col**. An example is:

COL (EMP-ID)

At validation time, ASF checks to see if catalog entries actually exist and checks the format of the WHERE-CLAUSE.

### Validate Once Only

At validation time, ASF validates *all* rows of the SRT that the user has access to. Due to this, you should wait until you have made all entries and changes to the SRT for all tables before validating anything. Then you need only perform the validation once in order for all these entries to be validated.

## Correcting Errors

If any of the catalog entries (user, group, owner) that have been entered don't exist or the format of the WHERE criteria is incorrect, the system displays the message: **ERROR ON VALIDATION, PROCEED TO MESSAGE SCREEN**. If this happens, identify and correct the error by performing the steps described below.

1. Press [PF5] to view the error messages on the Message screen. For example, if you tried to enter a group ID that was not entered in the catalog, you would get the message:

```

MSG5  CA - Automatic System Facility          CA          ** Message Screen **
                                           PF2 - Print Messages Page: 1 of 1
DC432065 (E) GROUP C1500, NOT DEFINED IN CATALOG
    
```

2. Press [Clear] to return to the RLOD screen. At the RLOD screen, press [Clear] again to return to the ASEL screen. On the ASEL screen, type RSEL in the screen name field. Press [Enter] to access the RSEL screen.
3. On the RSEL screen, next to the STATUS column, enter NE 'V' to select entries with validation errors:

```

RSEL  CA - Automatic System Facility          CA          *Row Level Security Selection*
DC560603 ENTER SELECTION CRITERIA
Table Name:
      _ PF1 - Help      _ PF2 - Message Screen      Page: 1 of 1

  ID      Column Name      Selection Criteria
0001 GROUP
0002 USER
0003 OWNER
0004 SECURITY-NAME
0005 STATUS                ne 'v'
0006 WHERE-CLAUSE
0007 GROUP-SEQUENCE-NUMBER
0008 TIME-STAMP
0009 DATE-STAMP
0010 WORK-GROUP

Additional Selection Criteria
    
```

4. Press [Enter] twice to display the invalid SRT entries on the RLOD screen:

CA	
PF1 = ADD;PF2 = CHANGE;PF3 = DELETE;PF4 = VALIDATE;ENTER = NEXT;CLEAR = EXIT	
\$SECURITY-RUNTIME-TABLE\$	
GROUP	C1500
USER	
OWNER	SNELL
SECURITY-NAME	SALSEC
STATUS	(E) ERROR IN VALIDATION
WHERE-CLAUSE	SALARY LT 1

5. Enter the correct information on the RLOD screen. Press [PF2] to change the row and then [PF4] to validate.

#### Assigning Security to a Group

If CA-ICMS is installed, you can associate security names with entire groups of users at once by associating the group ID (instead of the user ID) with a security name.

### Step 3: Associating Security Names with Groups

Assigning group IDs allows Fran Snell to define a single SRT entry for the group instead of defining individual entries for each user in the group. This would allow her to limit access to Forrest and White instead of just Forrest.

At validation time, the system automatically creates SRT entries for each member of the primary group but not for members of subgroups within the group. In order to associate a group ID with the security name for the APPLICANT table, follow the same steps used to assign a security name and WHERE criteria to an individual user. The only difference is that you specify a group ID in the GROUP field rather than a user ID in the USER field.

## Considerations

- When you initially validate an SRT entry with a group ID, ASF stores the following fields for each member of the group *at validation*:
  - OWNER
  - SECURITY-NAME
  - WHERE-CLAUSE
- If you change the SRT entry and a group ID is specified, ASF changes the following fields, for each member of the group, *after validation*:
  - OWNER
  - SECURITY-NAME
  - WHERE-CLAUSE

In either case, when fields are initially stored and when they are changed, if validation fails, then no fields are stored or changed.

### Disassociating Individual from Group

- When you modify an individual SRT entry that was generated as a result of a group entry, the individual entry is disassociated from the group as defined to the `SECURITY-RUNTIME-TABLE`. If you later modify the group entry, the individual entry is *not* modified.
- When you modify a group entry, the group field cannot be changed.
- When you delete a group entry, all the associated member entries that were generated at validation time are also deleted from the SRT.

## Step 4: Assigning Generic Names to Columns

In addition to associating a security name with more than one user, you can also assign a single security name to more than one table. This allows you to create a single SRT entry for tables with the same restrictions.

In the `JOB` table, Fran has information about the type of job under the column name `TITLE`. In the `APPLICANT` table, Fran has similar information about the type of job under the column name `POSITION APPLIED FOR`. Ideally, she would like to apply row-level security to both tables so that Forrest and White will see only the job positions that apply to their department (Office Services).

### WHERE Criteria Apply to One Column Name

To assign the same row-level security restrictions to both columns (`TITLE` and `POSITION APPLIED FOR`) in both tables (`JOB` and `APPLICANT`), you must be able to assign `WHERE` criteria to both of the columns. In the `WHERE` criteria you can name only one column at a time, for instance, `WHERE TITLE EQ 'OFFICE SERVICES'`.

Assigning a generic name to both columns allows you to use one name to refer to both columns. To assign a generic column name to the TITLE column in the JOB table and the POSITION APPLIED FOR column in the APPLICANT table, perform the following steps:

1. On the ASEL screen, select the JOB table.
  - Type RSEC in the screen name field and press [Enter] to access the RSEC screen.
  - Enter the security name of the table (SALSEC) in the Security Name field.
  - Enter the generic column name (POSITION) beside the TITLE column and press [Enter]:

```

RSEC  CA - Automatic System Facility          CA          ** Row Level Security **
DC560314 ADD/MODIFY SECURITY NAME AND GENERIC COLUMN NAMES
Table Name: JOB
                                           _ PFl - Help Page: 1 of 1

Security Name: salsec

          Column Name          Generic Column Name
JOB NUMBER
DEPARTMENT
TITLE                position
SALARY
CURRENT STATUS (FILLED OR OPEN)
    
```

**Note:** The generic column name cannot contain special characters.

- Generate the table.
1. On the ASEL screen, select the APPLICANT table.
    - Type RSEC in the screen name field and press [Enter] to access the RSEC screen:

```

RSEC  CA - Automatic System Facility          CA          ** Row Level Security **
DC560314 ADD/MODIFY SECURITY NAME AND GENERIC COLUMN NAMES
Table Name: APPLICANT
                                           _ PFl - Help Page: 1 of 1

Security Name: SALSEC

          Column Name          Generic Column Name
NAME
STREET
CITY
STATE
ZIP CODE
PHONE NUMBER
POSITION APPLIED FOR
SALARY REQUIREMENT
EVALUATION: STRONG, WEAK, MEDIUM
    
```

- Enter the generic column name (POSITION) beside the POSITION APPLIED FOR column and press [Enter].
  - Generate the table.
1. Make an SRT entry associating the security name with the WHERE criteria:
    - Type RLOD in the screen name field.
    - Specify the following information:
      - **Group**—The group ID of White's group (C1300)
      - **Owner**—Snell's user ID (SNELL)
      - **Security Name**—The single security name assigned to both of the tables (SALSEC)
      - **WHERE criteria**—The appropriate criteria using the generic column name assigned to the columns in each table (POSITION)
    - Press [PF1] to add the row.
    - Validate the entry by pressing [PF4]. The validation process checks the format of the information.

If the message **ERROR ON VALIDATION, PROCEED TO MESSAGE SCREEN** occurs, correct it by performing the steps listed in "Correcting Errors" discussed earlier in this section.

**Note:** Remember, assigning row-level security changes the definition of the table. In order to access the table, you must regenerate it.

## Considerations

When assigning security names to multiple tables, make sure the criteria apply only to the columns you intend them to. For example:

### Distinguishing Columns

**When two columns in different tables have the same name** and you want to apply WHERE criteria to only one of the columns, do one of the following:

- Change the name of one of the columns, preventing the WHERE criteria from being applied to both columns.
- Change the security name of one table, so the WHERE criteria will not apply to both tables.
- Assign a generic column name to one of the tables. This changes the name of that column.

### Grouping Columns

**When two columns in different tables have different names** and you want to use one SRT entry to apply the same WHERE criteria to both columns, assign a generic column name to the columns. When WHERE criteria are written using generic column names, the system regards the two columns as one and applies criteria to both. Generic column names must be alphanumeric character strings that do not contain hyphens or special characters.

## Row-level Security Summary

Row-level security offers you the ability to secure individual rows of data. With row-level security, if authorized to do so by the ASF administrator, you can decide which rows of a stored table a user is allowed to see.

The row-level security functions of ASF allow you to perform the following activities:

- Assign row-level security to a table by associating a security name with the table.
- Allow a user access to certain rows of a stored table by associating a user and WHERE criteria with a security name.
- Apply the same row-level security to multiple tables by assigning the same security name to the tables.
- Apply the same row-level security to columns in different tables by assigning a generic name to the columns



Assigning row-level security to a table changes the table definition. In order to access a table with row-level security, the table must be generated or regenerated after the row-level security has been applied.

Row-level security applies to other products, as follows:

- The Information Center Management System (CA-ICMS)—Supported
- Automatic System Facility (ASF)—Supported
- OnLine Query (CA OLQ)—Supported
- CA Culprit—Supported
- Application Development System (CA ADS)—Not supported
- Device Media Language (DML)—Not supported

**Note:** None of the tools mentioned above will respect the row-level security defined to records that are used in any subschema except the RU subschema.



# Chapter 8: ASF Processing

---

In its processing, ASF uses a number of other products and facilities, including:

- The **data dictionary** (IDD)
- The **catalog** structure within the data dictionary
- The **schema and subschema compilers** of CA IDMS/DB
- The **Logical Record Facility** (LRF)
- The **CA ADS dialog compiler**
- The **CA IDMS/DC and CA IDMS UCF mapping compiler**

This chapter is intended for those (most likely programmers and DBAs) who need or want to know how ASF works in order to use ASF's extended and administrative functions effectively. The following topics are discussed:

- How data tables are defined
- How data tables are stored
- How data tables are generated

This section contains the following topics:

[Table Definition](#) (see page 115)

[Table Storage](#) (see page 118)

[Table Generation](#) (see page 119)

## Table Definition

### Definition Records Defined

Tables created through ASF are defined in a system-supplied schema named **IDMSR**. The schema IDMSR is provided at installation. When installed, it contains the definitions of **definition records**, which are used to store table definitions. The following definition records are defined in the system-supplied IDMSR schema:

- **RDEFREC**—Defines a table.
- **FDEFREC**—Defines the external column characteristics of a stored table.
- **EDEFREC**—Specifies the internal column (field) characteristics of a stored table.
- **RDERREC**—Specifies derivation information. This record is used to connect columns in views to the tables from which the views were derived. Additionally, this record specifies if the view is derived from a stored table or a stored non-SQL defined record.

- **NDERREC**—Identifies a table definition that was derived from a stored record.
- **JOINREC**—Specifies the information necessary to connect two RDERRECs together with join criteria. This construct provides the information necessary to build a logical-record path for accessing the source records.
- **RSELREC**—Saves selection criteria during table definition.
- **WHERREC**—Contains selection criteria used to build logical-record path DML commands.
- **VALUREC**—Contains calculations defined for columns in the table.
- **KEYHREC**—Defines an index key for a table.
- **KEYEREC**—Defines the elements in an index key.
- **GCTLREC**—Contains ASF system defaults and control information.
- **AREAREC**—Defines an area managed by ASF.
- **RIDSREC**—Contains a list of available record IDs.
- **RDNSREC**—Contains a list of available table definition numbers.
- **ORGNREC**—Defines the characteristics of a stored table as it relates to a logical record.

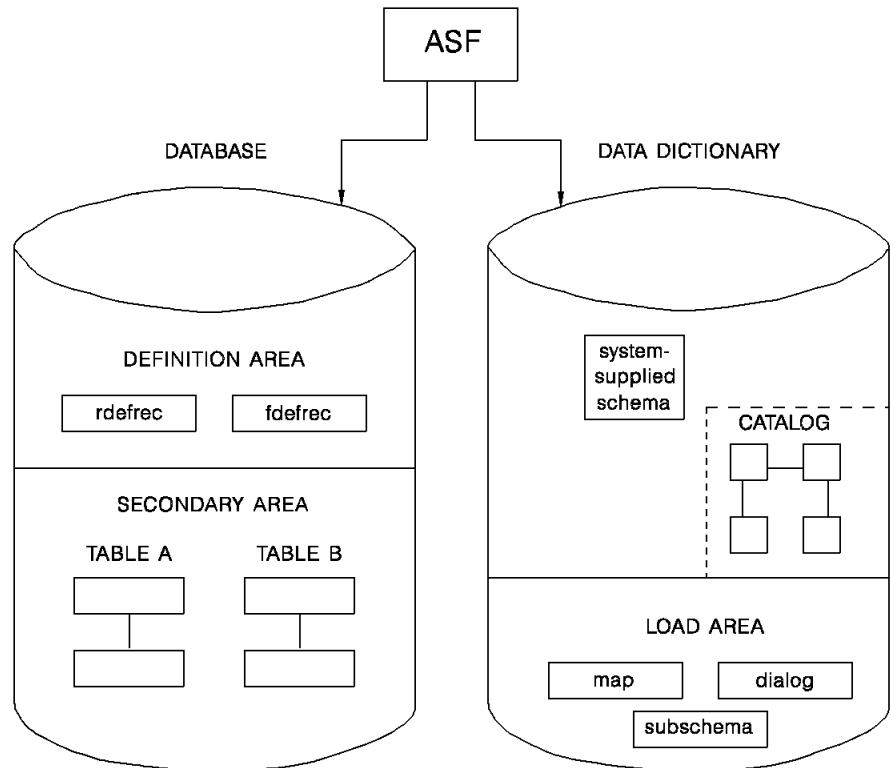
### Definition Records Have Owners

Each definition record has a corresponding **owner record**. The first four characters of the owner record name are the same as the first four characters of the member record name. The last three characters of each owner record name are **OWN**. For example, the owner of the RDEFREC record is named RDEFOWN.

The definition records and their owners are stored in the definition area of the database.

### User and Table Catalog

The **catalog**, which is part of the data dictionary, is a directory of users and tables. ASF references and updates the catalog as it defines data tables. In CA-ICMS installations, ASF and CA-ICMS share the same catalog. The figure below illustrates the dictionary and database components required by ASF.



### Defining Tables

As you go through the table definition process, the following steps take place:

1. **When you enter a table name** on the Activity Selection screen, ASF checks the catalog to see if the table exists. If the table does not exist, ASF instructs you to proceed with the table definition.

2. **When you press [Enter]** from the Table Definition screen, ASF adds the table. Selecting a function, such as Define Columns, also causes the table to be added. ASF adds the table as follows:
  - ASF assigns the table a table definition number.
  - Using the table definition number to identify the table, ASF adds an occurrence of the Record Definition record (RDEFREC).
  - The table name and definition number are entered in the catalog.
3. **As you define the characteristics of the table**, ASF adds occurrences to the appropriate definition records. For example:
  - One FDEFREC occurrence is added for each column in the table.
  - One KEYHREC occurrence is added for each index key defined for the table.

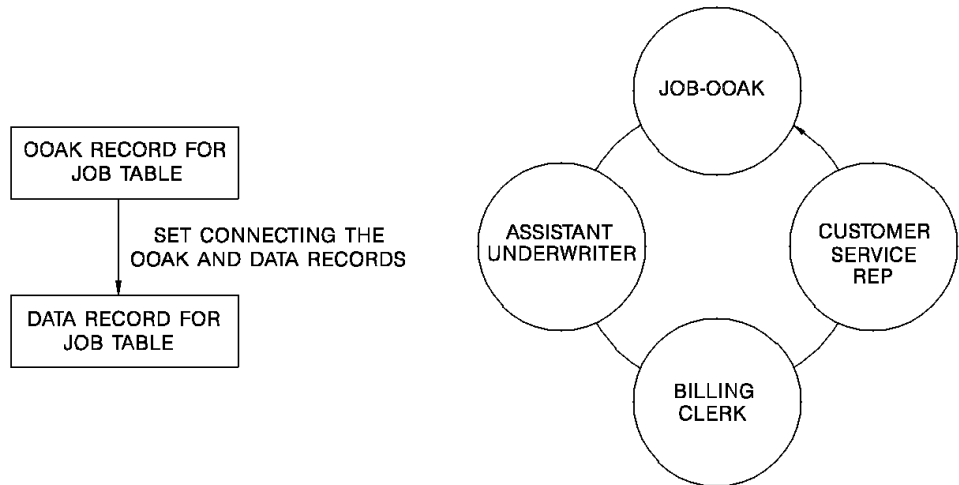
The definition records store the table specifications you enter through ASF.

## Table Storage

Each stored table has the following structure in the database:

- An OOA (one-of-a-kind) owner record
- A DATA member record, which is defined using the specifications entered by the user
- At least one indexed set that connects the OOA record and the DATA member record

The rows of a table are stored as occurrences of the DATA record. The table is accessed by walking the indexed set that connects the OOA and DATA records and obtaining all DATA records in the set. The figure below illustrates the internal table structure.



## Table Generation

When you generate a table through ASF, the following steps take place:

1. **Record and element** definitions are added to the data dictionary. The naming conventions are as follows:
  - Records are named RFUR-*nnnnnn*-DATA, where *nnnnnn* is the table definition number.
  - Element names are the Stored Column Name shown on the Extended Column Definition screen.
2. **Record and set** descriptions are added to the schema IDMSR (*nnnnnn* is the table definition number) as shown in the following table.

Name	Object	Description
RFUR- <i>nnnnnn</i> -DATA	record	The record type in which rows of the table are stored as record occurrences.
RFUR- <i>nnnnnn</i> -OOAK	record	The CA IDMS/DB-defined OOAK record.
RFUS- <i>nnnnnn</i> -SET	set	An unsorted indexed set defined if the storage sequence of the table is NEXT, PRIOR, FIRST, or LAST.
RFPS- <i>nnnnnn</i> -SET	set	An indexed set defined if the storage sequence of the table is designated by the key number of an index key.
RFFS- <i>nnnnnn</i> -yyyy	set	One for each fast access key defined on the Extended Key Definition screen (refer to the chapter "Extended Functions of ASF"). Yyyy refers to the field definition number for the column, assigned in the FDEFREC.
RFKS- <i>nnnnnn</i> -zzzz	set	One for each additional key defined on the Extended Key Definition screen that is not used as the storage sequence. Zzzz is the key number of the additional key.

3. A **subschema** (named RUnnnnnn) is generated consisting of:
  - A **logical record** containing the elements shown in the following table.

Name	Description
ASF-RDEF-REC	The table definition record
RFUR- <i>nnnnnn</i> -OOAK	The system-defined OOAK record

Name	Description
RFUR- <i>nnnnnn</i> -DATA	The record used to store rows of the table
RFUR- <i>nnnnnn</i> -WORK	An IDD work record, included if any work fields are defined for the table

The logical record is named by modifying the table name to conform to COBOL naming conventions, as shown below.

Special characters and spaces are changed to hyphens:

- JOBS AND APPLICANTS becomes JOBS-AND-APPLICANTS
- \$\$SALARY\$\$ becomes --SALARY--

Leading and trailing hyphens are deleted:

- --SALARY—becomes SALARY

Lowercase letters are translated to uppercase letters:

- phone list becomes PHONE-LIST

Multiple repeating embedded hyphens are reduced to one hyphen:

- EMPLOYEES----NEW becomes EMPLOYEES-NEW

Numbers in the first position are preceded by an R:

- 1985 BUDGET becomes R1985-BUDGET

Names longer than 16 characters are truncated:

JOBS AND APPLICANTS becomes JOBS-AND-APPLICA

- **Logical-record paths**, which allow an application program to OBTAIN, STORE, MODIFY, and ERASE occurrences of the logical record.
- 4. A **map** is created in which the columns defined for the table are the data fields for the map. The map is named RM*nnnnnn* (*nnnnnn* is the table definition number). This step can be suppressed at the user's discretion. If you suppress this step, step 5 is also suppressed.
- 5. A **dialog** that uses the subschema and map is generated to allow online access of the table through ASF. The dialog is named RD*nnnnnn* (*nnnnnn* is the table definition number). In addition to the subschema (RU*nnnnnn*) and map (RM*nnnnnn*), the dialog has the following components:
  - **ASF-PARM-REC**—A parameter record included in every ASF-generated dialog
  - The following **Process modules**.

Name	Description
RD <i>nnnnnn</i> -PREMAP	The premap process
RD <i>nnnnnn</i> -CLEAR	The response process for the [Clear] key



---

<b>Name</b>	<b>Description</b>
RDnnnnnn-ENTER	The response process for the [Enter] key
RDnnnnnn-PA1	The response process for the [PA1] key
RDnnnnnn-PF1	The response process for the [PF1] key
RDnnnnnn-PF2	The response process for the [PF2] key
RDnnnnnn-PF3	The response process for the [PF3] key

The first two steps above apply only to stored tables. For views, ASF performs steps 3 through 5 only.



# Chapter 9: Extended Functions of ASF

---

The basic functions of ASF enable end users to develop applications that answer some of their data processing needs. ASF also provides programmers and data processing professionals with **extended functions**, which enable them to perform the following activities:

- Specify underlying table characteristics.
- Define column attributes.
- Select data at definition time.
- Derive views from various sources.
- Create indexes for a table.

This section contains the following topics:

[Specifying Underlying Table Characteristics](#) (see page 123)

[Defining Column Attributes](#) (see page 127)

[Selecting Data at Definition Time](#) (see page 137)

[Deriving Views from Various Sources](#) (see page 139)

[Creating Indexes](#) (see page 150)

## Specifying Underlying Table Characteristics

Each table defined through ASF includes underlying characteristics automatically supplied by ASF. You can use the **Extended Table Definition** screen to control aspects of the table definition process usually handled by ASF.

A sample Extended Table Definition screen is shown below.

```

                                CA
XTAB  CA - Automatic System Facility      *Extended Table Definition*
DC560501 MODIFY TABLE DEFINITION
Table Name: JOB

  _ PF1 - Help                      _ PF4 - Extended Key Definition
  _ PF2 - Extended Table Derivation  _ PF5 - Row Level Security
  _ PF3 - Extended Selection Definition

Area: IDMSR-AREA2

Primary Number Of Rows...: 20  Subschema: RU000239
Secondary Number Of Rows: 20  Map.....: RM000239  Version: 1
Maximum Number Of Rows...: 40  Dialog...: RD000239  Version: 1

Display Access.....: YES      Table Definition Length: 120
Load Access.....: YES
Change Access.....: YES      Date/Time Created.....:mm/dd/yy 10:20:4
Erase Access.....: YES      Date/Time Last Updated.:mm/dd/yy 10:50:3

Generate Map/Dialog.....: YES  Process Standard Area Data: YES

C/ICMS Modify Option....: YES  C/ICMS Selective Update: YES

Logical Record Name.....: JOB
    
```

**Extended Table Definition Functions**

The Extended Table Definition screen can be used to:

- Assign a table to an area other than the default area automatically assigned by ASF, or move the table to a new area.
- Specify how space in the database is to be allocated for storing rows of the table.
- Limit the use of the table, both through ASF and outside ASF.

**Assigning a Table to an Area**

A table is assigned to an area in one of several ways.

**Assigned by Administrator**

An ASF administrator can assign a table to a particular area by entering the area name in the Area field. The area must be a valid area that has been identified to ASF through the Add Area function of the Administrator Control screen (see the chapter "Administrative Functions of ASF"). The Area field is protected if the user does not have administrative authority.

**Default Assignment**

The table is assigned to the default area designated for the user defining the table. On the User Limits and Defaults screen, an ASF administrator can assign a default area for all of a user's tables (see the chapter "Administrative Functions of ASF").

**System Default**

If no default area has been assigned for the user, ASF assigns the table to the default area for the system, as specified on the Administrator Control screen (see the chapter "Administrative Screens").

**Changing the Area**

The area name is displayed in the Area field when the table is defined.

Once the table is generated, an ASF administrator can assign the table to a new area by changing the name in the Area field. When the table is regenerated, the table's data is moved to the new area.

## Storing a Table in an Area

**A Table Goes in one Area**

Space in an area is preallocated. The area should, therefore, be large enough to accommodate the *maximum* number of rows that will be stored for the table. If you know in advance that a data table will be very large, you can create an area for the table before you load any of the table rows.

When storing a table in an area, you have the following options available on the Extended Table Definition screen:

- Choosing whether or not to restructure the table at regeneration
- Choosing whether or not to delete the table when deleting the table definition

**MAINTAIN ASF Utility**

Tables can be loaded through MAINTAIN ASF, the table load utility. This utility:

- Provides a fast load capability for large tables with multiple indexes
- Supports definition-only table redefinition, which allows you to redefine the table without converting the data

- Supports definition-only table deletion, which allows you to delete the table definition without erasing the data

To achieve optimal use of MAINTAIN ASF, you should store only one table per area. This way, you can simply reinitialize the area when you want to delete table data. For more information on MAINTAIN ASF, see the chapter "MAINTAIN ASF".

## Limiting the Use of a Table

By default, when you generate a table through ASF, an online application is created that enables you to display, add, change, and delete data in the table. You can choose not to generate an application for the table, or you can limit the functions that can be performed by both the ASF-generated application and any other applications that use the table.

### Prevent Table Generation

To instruct ASF not to generate an application for the table, change the Generate Map/Dialog field from YES (default) to NO. The table will not be available for use through the Load Data, Display/Change Data, and Select Data functions of ASF. The table can be loaded and updated through batch or online programs developed outside ASF, as well as through OnLine Query (CA OLQ) and CA Culprit.

**Note:** The Generate Map/Dialog field can also be used to generate an application for a table created outside ASF (for example, through OnLine Query or CA Culprit). Change the NO (the default for tables created outside ASF) to YES and regenerate the table.

### Restrict Table Use

To restrict the functions that can be performed with the table, both through the ASF-generated application and applications developed outside ASF, use the following fields:

- **Display Access**— Enter NO to prevent users from displaying the table. The OBTAIN logical-record path is omitted from the generated subschema.
- **Load Access**— Enter NO to prevent users from loading data into the table. The STORE logical-record path is omitted from the generated subschema.
- **Change Access**— Enter NO to prevent users from modifying the table. The MODIFY logical-record path is omitted from the generated subschema.
- **Erase Access**— Enter NO to prevent users from deleting data from the table. The ERASE logical-record path is omitted from the generated subschema.

Since the above restrictions apply to the generated subschema, any applications that use that subschema, such as a DML program, are also restricted.

The remaining fields on the Extended Table Definition screen give information about a generated table and application.

For more information about fields on the Extended Table Definition screen, see the chapter "Extended Screens"

## Defining Column Attributes

The **Extended Column Definition** screen, shown below, enables you to specify additional characteristics for a column defined on the Column Definition screen and to change the default values assigned for the column by ASF, as follows:

- Specify the storage and display format of a column.
- Define an initial value or calculation for a column.
- Change the mapping attributes automatically defined for a column.

Each of these topics is discussed separately below.

XCOL CA - Automatic System Facility		CA	
DC560407 MODIFY COLUMN DEFINITION		*Extended Column Definition*	
Table Name: EMPLOYEE			
_ PF1 - Help		_ PF2 - Next Column	
_ PF3 - Message Screen			
Displayed Column Name		Stored Column Name	
DEPT-ID-0410		DEPT-ID-0410	
Display Format		Stored Format	
Z(0004)		9(0004)	
Usage: D D/0/1/2/3 Display/Comp/Comp-1/Comp-2/Comp-3			
Comments.....:		Column Level: ELEMENT	
Initial Value:			
Calculation..:			
Edit Table:	Version:	Code Table:	Version:
U U/P	Unprotected/Protected	N B/R/P/T/G/Y/W/N	Blue/Red/Pink/Turquoise
Green/Yellow/White/Nocolor			
N Y/N	Blank When Zero	N N/R	Normal/Reverse Video
D B/D/K	Bright/Display/Dark	N Y/N	Blink/NoBlink

## Specifying Column Formats

ASF generates an internal and external picture for each column you define, based on the type and width specified on the Column Definition screen. The external picture is displayed in the Display Format field of the Extended Column Definition screen; the internal picture is displayed in the Stored Format field. The following table shows the default formats for each column type.

### Default Display and Storage Formats

Column type	Display format (external picture)	Storage format (internal picture)
TEXT	X(n)	X(n)
NUMERIC	-Z(n)	S9(n)
CURRENCY	-Z(n-2).99	S9(n-2)V99

**Note:** The length of text fields is limited only by the size of the database page. The lengths of numeric fields and currency fields are 18 characters.

### Change Defaults

You can change the default format provided:

- The Display Format can be modified for all table types (stored or view).
- The Stored Format can be modified for stored tables only.

**Note:** The Width on the Column Definition screen corresponds to the Stored Format (internal picture). The Display Format (external picture) determines the size of the response field on the generated map.

### Specify New Formats

Specify the display and storage formats, as follows:

- **Alphanumeric data** is described by using the following characters:
  - **X**— The character X represents one alphanumeric character.
  - **(n)**— An integer in parentheses after an X represents *n* repetitions of the alphanumeric character; for example, X(4) is equivalent to XXXX.
- **Alphabetic data** is described by using the following characters:
  - **A**— The character A represents one alphabetic character (A-Z, a-z).
  - **(n)**— An integer in parentheses after an A represents *n* repetitions of the alphabetic character.



- **Numeric data** is described by using the following characters:
  - **9**— The character 9 represents one numeric character.
  - **(n)**— An integer in parentheses after a 9 represents *n* repetitions of the numeric character.
  - **V**— The character V represents an assumed decimal point. No more than one V can appear in a column format. If the V is omitted and the P option (described below) is not used, the assumed decimal point is after the rightmost 9.
  - **P**— The character P represents an assumed zero. Any number of Ps can appear in the leftmost or the rightmost positions of a column format. An assumed decimal point is automatically placed before the first P or after the last P.
  - **S**— The character S identifies the number as signed, positive or negative. When used, the S must be the first character in the column format. When the S is omitted, values for the column are assumed to be positive.
- **Numeric edited data** is described by using the characters described above for numeric data, along with the following editing characters:
  - **Z**— A zero-suppression indicator is replaced from right to left with the corresponding digit unless the digit is a leading zero. A Z replaces any zeros to the left of the first nonzero digit with blanks.
  - **\$**— A dollar sign indicator outputs a dollar sign in the position it occupies in the edit mask.
  - **+**— A sign indicator is always included in the picture.
  - **-**— A minus sign indicator is included in the picture when the number is negative. No sign is included if the number is positive.
  - **.**— A decimal point indicator is included in the picture in the position it occupies in the edit mask.
  - **,**— A comma outputs a comma in the position it occupies in the edit mask.

These characters represent edit symbols and can be used only for the display format of a column.
- If Double Byte Character Set support (DBCS) is installed, **Graphic display format** may be specified to describe graphic data using the following characters:
  - **G**— The character G represents one graphic (DBCS) character.
  - **(n)**— An integer in parentheses after a G represents *n* repetitions of the graphic (DBCS) character.

### Specify Usage

The values specified in the Usage field can affect the storage and display formats of a column. By default, the column usage is DISPLAY for text columns and COMP-3 (packed decimal) for numeric columns. The default display and storage formats shown in :refid=formats. above apply to the default usage. If the usage COMP (binary) is specified for a numeric column, the default display and storage formats also apply.

### Floating Point Numbers

ASF also supports floating point numbers, both short (single-precision) and long (double-precision). If a usage of COMP-1 (short floating point) or COMP-2 (long floating point) is specified, the display and storage formats are altered as follows:

- The Stored Format field is erased because no internal picture applies to floating point numbers. Either a full word (COMP-1) or a double word (COMP-2) of storage is allocated for the column.
- The display format determines the size of the map field associated with the column in the ASF-generated application. The original default in the Display Format field remains unchanged when the usage is modified. It is recommended that you erase the display format currently displayed and allow it to default to the appropriate display format for the usage specified. This provides a map field large enough for the maximum number that can be stored.
- The Width field on the Column Definition screen is automatically modified to reflect the usage specified. The width is 4 for COMP-1 and 8 for COMP-2.

## Defining Column Levels and Values

### Three Levels of Columns

A column can be one of three column levels, as follows:

- **Element** is the default for columns in stored tables and for most derived columns in views. Element-level column entries are stored in the database.
- **Work** is the default for columns added to views. Work columns are for display purposes only, and are usually defined by a calculation or participate in a calculation elsewhere in the table. A column in a stored table can be defined as a work column.

**Note:** You cannot apply selection criteria to work columns. You can, however, apply the criteria to the columns involved in the calculation that results in the work column.

- **Group** can apply only to columns that are derived from non-SQL defined records. Group-level columns do not contain data themselves, but serve as headings for subordinate elements. You cannot delete a group-level column; you can exclude the column from the screen display.

### Predefined Column Values

For element-level and work-level columns, you can include a value for the column as part of the column definition. Column values can be predefined in various ways.

#### Initial Value

You can specify an initial value that provides a default value for any column except one with a display format of **G**. You can define an initial value for a column that is appropriate for most rows in the table. The initial value must be consistent with the column width and type, and can be a number or a literal (literals must be enclosed in quotation marks). If no initial value is specified, ASF provides the following initial values:

- Blank for alphanumeric columns
- Zero for numeric columns

#### Calculated Value

You can specify a calculation for a column that is computed when the column is displayed or stored.

You can define both an initial value and a calculation for a single column, but most often you will use one or the other.

The initial value for a column appears as the column entry when you display the table to add a row. If no other value is supplied for the column when the table is stored, the initial value is stored as the column entry.

A calculation is an expression that is evaluated by ASF, as follows:

- If the calculation is for an element-level column, the expression is evaluated when the row is stored or modified.
- If the calculation is for a work-level column, the expression is evaluated when the row is displayed and is reevaluated when the row is modified. When adding rows, the expression is evaluated when a row is added. The evaluation is for display purposes or for calculating intermediate results only, since the entry in a work column is not actually stored.

### Fixed Value

A calculation can be a single value, like an initial value, that ensures a fixed value for a column. The value specified as the calculation overrides any value entered in the column and is always the value displayed or stored. The value specified can be a number, literal (enclosed in quotation marks), or another column in the table. The most common use of a single-value calculation is to reference another column in the table, as shown in Example 2 below.

Though single-value calculations are allowed, a calculation is typically an **arithmetic expression** that is calculated for a numeric column or for a text column containing numeric data.

As noted above, a calculation overrides any value entered in a column, though an entered value can be used to evaluate the expression in the calculation.

Arithmetic expressions can include:

- Numeric values
- References to one or more columns in the table, including the column for which the calculation is being defined. References can be designated:
  - By entering the column name
  - If the column name includes special characters, by entering `COL(column-name)`
  - By entering the ID assigned to the column on the Extended Selection Definition screen, preceded by a pound sign: `#column-id`
- Arithmetic operators:
  - `+—` Addition
  - `-—` Subtraction
  - `*—` Multiplication
  - `/—` Division
- Parentheses to separate operations

ASF follows the standard order of precedence when evaluating arithmetic expressions. The order is: multiplication or division from left to right, then addition and subtraction from left to right, with operations in parentheses resolved first. If more than one calculation has been defined in a table, the calculations are evaluated in the order in which the columns were defined.

## Examples

The following examples illustrate how initial values and calculations can be used in applications developed by ASF.

### Defining an Initial Value for a Column

The initial value for the HOURS column is:

INITIAL VALUE: 40

The EMPLOYEE (OT) table is displayed through the Load function:

```
EMPLOYEE (OT)
NAME
NUMBER
GRADE
HOURS                40
WAGE
REGULAR PAY
OVERTIME PAY
TOTAL PAY
```

### Defining a Single-Value Calculation

The EMPLOYEE table and the POSITION table are joined on the NUMBER and EMPLOYEE NUMBER columns. The EMPLOYEE NUMBER column is not displayed in the EMPLOYEES AND THEIR POSITIONS view. To ensure that the join columns are always equal, a calculation is defined for the EMPLOYEE NUMBER column, as follows:

CALCULATION: NUMBER

When the NUMBER column of the EMPLOYEE table is modified through the view, the EMPLOYEE NUMBER column is automatically modified in the POSITION table. The modification is allowed through the view because both join columns are modified and the join condition remains true.

### Defining an Arithmetic Calculation for a Column

The calculation defined for the OVERTIME PAY column is:

CALCULATION: (HOURS - 40) \* (WAGE + (WAGE \* .5))

The result displayed in the EMPLOYEE (OT) table is:

EMPLOYEE (OT)	
NAME	SAM MORRIS
NUMBER	4236715B
GRADE	C
HOURS	50
WAGE	12.75
REGULAR PAY	
OVERTIME PAY	191.25
TOTAL PAY	

### Defining Calculations for More than One Column

The calculation for the OVERTIME PAY column is:

CALCULATION: (HOURS - 40) \* (WAGE + (WAGE \* .5))

The calculation for the REGULAR PAY column is:

CALCULATION: 40 \* WAGE

The calculation for the TOTAL PAY column is:

CALCULATION: REGULAR PAY + OVERTIME PAY

The results displayed in the EMPLOYEE (OT) table are:

EMPLOYEE (OT)	
NAME	SAM MORRIS
NUMBER	4236715B
GRADE	C
HOURS	50
WAGE	12.75
REGULAR PAY	510.00
OVERTIME PAY	191.25
TOTAL PAY	701.25

## Mapping Considerations

The bottom third of the Extended Column Definition screen defines the following map characteristics for the online application generated by ASF:

- Automatic editing criteria for a column
- Map attributes for a column

Each of these topics is discussed separately.

## Defining Automatic Editing Criteria

Automatic editing can be used to edit and validate data being entered into a table. The following criteria can be defined for automatic editing:

### 1. Edit Table

An edit table defines a set of valid or invalid values or ranges of values for a column. The edit table is used to validate data added to the column. An edit table can contain either valid or invalid values:

- **If the edit table contains valid values**, data input in the column is valid only if it is listed in the table.
- **If the edit table contains invalid values**, data input in the column is valid only if it is not listed in the table.

### 2. Code Table

A code table defines values for encoding and decoding data. The code table is used to convert data typed in a column to its encoded form for storage and to decode the stored data for display.

### Define Edit, Code Tables with IDD

Both edit and code tables must be defined through IDD. You can identify an edit or code table to ASF for a particular column by entering the table names in the **Edit Table** and/or **Code Table** field of the Extended Column Definition screen.

## Defining Map Attributes

### Screen Display

An **attribute** is a characteristic provided by a terminal. Different characteristics can be assigned to fields on a map. For example, the BRIGHT attribute can be assigned to fields that should be displayed at a brighter-than-normal intensity. The BLINK attribute can be assigned to fields that should blink at run time.

You can specify the map attributes for a column on the Extended Column Definition screen. If you do not specify attributes for a column, default attributes are used for the column.

The attributes for a column take effect when the table is displayed on the screen at run time. The following table describes the attributes typically provided by 3270- and 3279-type terminals.

### Attributes Provided by 3270- and 3279-type Terminals

Terminal	Attribute	Description
3270	PROTECTED/ UNPROTECTED	A PROTECTED column will not accept data from a terminal operator. An UNPROTECTED column will accept data from a terminal operator.
3270	DISPLAY/BRIGHT/ DARK	DISPLAY specifies that the column entry will appear on the screen at normal intensity. BRIGHT specifies that the column entry will appear at high intensity. DARK specifies that the contents of a field are not visible on the screen at run time.
3279	BLINK/NOBLINK	BLINK specifies that the column will blink at run time. NOBLINK specifies that the column will not blink.
3279	NORMAL VIDEO/ REVERSE VIDEO	NORMAL VIDEO specifies that the color of the column and of the background are not reversed. REVERSE VIDEO specifies that the colors are reversed.
3279	BLUE/RED/PINK/ GREEN/TURQUOISE/ YELLOW/WHITE/ NOCOLOR	Any one of these color attributes can be assigned to a column. NOCOLOR specifies that the default display color for the terminal will be used at run time.



ASF defines the following default attributes for columns:

- UNPROTECTED
- DISPLAY
- NOBLINK
- NORMAL VIDEO
- NOCOLOR

REVERSE VIDEO and BLINK are mutually exclusive; they can be specified in conjunction with other attributes, but they cannot be specified with each other.

The **Blank When Zero** field, which is listed with the map attributes, indicates how a numeric column is to be displayed when the value of the column is zero. The default is N (zeros are displayed), but you can specify Y so that the column is blank when zero.

## Selecting Data at Definition Time

The **Extended Selection Definition** screen is used to define selection criteria for a table. The criteria specified are used to select the data to be included in the table. Selection definition is applied most often to views, but can be specified for stored tables. Defining selection criteria can serve the following purposes:

- **When defined for a view**, the criteria are used to select the source data to be included in the view.
- **When defined for a stored table or view**, the criteria are used to edit data added to the table.

### Use Edit Table for Editing

The use of selection definition for editing data is limited, particularly through the application generated by ASF. When a row being added does not meet the selection criteria defined, the message INVALID DATA is displayed, with no indication as to which column entry is in error. Edit tables are more effective and should be used when appropriate. The primary purpose of selection definition is to select rows for inclusion in a view.

### Defining Selection Criteria

Criteria for selection definition are specified in the same way as criteria for data selection at display time. Selection criteria are defined in the following two ways:

- **Fixed-format** expressions allow you to define selection criteria on a column-by-column basis, using simple logical expressions (including arithmetic expressions) that contain a single operand (such as EQ, GT, or MATCHES). With fixed-format expressions, you can define criteria for more than one column. The system then selects rows where all expressions are true (that is, using an implied AND as a conjunction between each fixed-format expression).
- **Freeform** expressions allow you to define selection criteria using compound logical expressions that contain several operands connected with AND, OR, and NOT. Freeform expressions can be used in addition to, or instead of, fixed-format expressions.
- **Freeform and fixed-format** expressions used together simply are treated as one expression with AND as the operator.

The Extended Selection Definition screen shown below illustrates the use of both fixed-format and freeform expressions to define selection criteria. For more information about rules on entering selection criteria, see the discussion on the Data Selection screen in the chapter "Basic Screens".

```
CA
XSEL CA - Automatic System Facility      *Extended Selection Definition*
DC560603 ENTER SELECTION CRITERIA
Table Name: EMPLOYEE1
      _ PF1 - Help      _ PF2 - Message Screen      Page: 1 of 1

ID      Column Name      Selection Criteria
0001 EMP-ID-0415
0002 EMP-FIRST-NAME-0415
0003 EMP-LAST-NAME-0415
0004 EMP-STREET-0415
0005 EMP-CITY-0415
0006 EMP-STATE-0415
0007 EMP-ZIP-0415      MATCHES '02###' 1
0008 EMP-PHONE-0415      MATCHES '617#####'
0009 EMP-STATUS-0415      '01'
0010 EMP-START-YEAR-0415      GE '1980'

Additional Selection Criteria

$7 OR $82
```

1. Fixed-format expression
2. Freeform expression

## Deriving Views from Various Sources

Using the Table Definition screen, you can derive a view from one or two existing data tables. With the **Extended Table Derivation** screen, you can:

- Derive a view from up to six sources
- Derive a view from non-SQL defined records

A sample Extended Table Derivation screen is shown below.

Column #1/Set	Of Der	Oper	Column #2	Of De
r1				
DEPT-EMPLOYEE2	023	ST4		0
5				

- 
- 1 Identifying the source tables

Joining the tables on an owner-member set relationship

- 2 Set name
- 3 Number of set owner
- 4 Join operator
- 5 Number of set member

The Extended Table Derivation screen is divided into two sections, as follows:

- The **top half** of the screen is used to identify source tables. This portion of the screen can be paged. A total of six sources can be identified for derivation.
- The **lower half** of the screen is used to specify join information.

### To Identify Sources

To identify sources for table derivation enter the appropriate information for each source, depending on its type:

- If the source is a stored record created by coding schema DDL, enter the Source Name and Subschema.
- If the source is a data table created through ASF, CA-ICMS, CA OLQ, or CA Culprit, enter the Source Name and Source Owner. The source table itself may be a derived view, so long as the source table is not derived from non-SQL defined records.

**Note:** Data tables and non-SQL defined records cannot be joined to each other. All derivation sources must be of the same type.

Page the top half of the screen forward ([PF8]) to identify additional sources for derivation. ASF numbers each source consecutively. The source table numbers are used to specify join criteria.

### Specify Join Criteria

When all the source tables have been identified, you can specify join criteria for a view in the lower half of the screen:

- Join data tables based on common column values
- Join non-SQL defined records on set relationships

When entering join information, you can page the top half of the screen to reference the numbers assigned to the source tables. Since the join criteria apply to the entire view, it does not matter which sources are displayed in the top half of the screen at any given time.

**Note:** When deriving a view from more than one source, every derivation source identified must participate in at least one join operation.

Use the Extended Table Derivation to:

- Derive views from data tables
- Derive views from non-SQL defined records

Each of these topics is discussed separately on the following pages.

## Deriving Views from Data Tables

When deriving views from data tables, the Extended Table Derivation screen is an extension of the derivation portion of the Table Definition screen. It enables you to identify additional source tables and join criteria.

The first two source tables identified and the first join condition specified will be displayed on both screens.

Use the Extended Table Derivation screen to derive from additional tables.

### Identify Sources

To identify tables for derivation, use the upper portion of the screen. Identify up to six source tables by completing the Source Name and Source Owner fields under each Derivation Source heading. Page forward as needed to identify additional source tables; a maximum of six source tables can be identified.

### Specify Join Criteria

To specify join criteria, move the cursor to the lower portion of the screen. Use the following steps to enter each join condition:

- **Type the name of the first join column** under Column #1/Set.
- **Specify the source table to which the column belongs** by typing the derivation source number under Of Der.
- **Enter the operator** on which the data in the two join columns is to be joined. The operators are EQ (equal), NE (not equal), GT (greater than), LT (less than), GE (greater than or equal to), and LE (less than or equal to). **Type the name of the second join column** under Column #2.
- **Specify the source table to which the column belongs** by typing the derivation source number under Of Der.

**Note:** When specifying the derivation source number for a join column, be sure to verify the number assigned to the source table in the upper portion of the screen.

### Join Considerations

When joining source tables, the following conditions apply:

- **Joining on a not equal (NE) condition** does not imply a no match condition. For example, if you join the CUSTOMER table to the ORDER table where the customer numbers are not equal, the view *will not* contain rows of customers without orders. It *will* contain rows of customers with every order but their own.

- **Reflexive joins** are allowed, enabling a table to be joined to itself. The table must be identified twice and referenced by two different derivation source numbers in the Of Der fields. It is recommended that the resulting view be used for display purposes only; updates are applied only to one of the rows in the source table rather than to the two rows displayed through the view.
- **Additional conditions** for joining two tables can be specified using the Extended Selection Definition screen.

A source table used to derive a view can be a view itself so long as the source was not non-SQL defined records.

## Deriving Views from Non-SQL Defined Records

Non-SQL defined records are defined by the DBA using schema DDL and the schema compiler.

### Extended Derivation Required

You must use the Extended Table Derivation screen to derive views from non-SQL defined records. The derivation portion of the Table Definition screen cannot be used for this type of derivation. The tables from which you can derive must be made available to ASF in the following ways:

### Define Records in IDMSR Schema

Their record descriptions must be included in the system-supplied schema, IDMSR, along with the appropriate related structures, such as area and set descriptions (see *CA IDMS Database Administration Guide* for details).

**Note:** It is recommended that any non-SQL defined records that are to be used in ASF be added to the schema IDMSR immediately after ASF system initialization (see the chapter "Administrative Functions of ASF") to prevent problems with duplicate record element names.

### Include Records in Subschema of IDMSR

They must be included in a subschema generated from the schema IDMSR that is accessible to you through dictionary security.

**How To Do It**

To derive views from non-SQL defined records:

1. Enter the **record name**, as defined in the schema, in the Source Name field in the upper portion of the screen. Specify the **subschema name** in which the record is defined in the Subschema field. Page forward as necessary to identify additional source records; a maximum of six source records can be identified. Each source record is numbered consecutively. The derivation source number is used to specify join information.

**Note:** All records used to define a single view must be defined in the same subschema. The subschema name must be specified for each source record identified.

2. Join the records on **owner-member set relationships** by entering the following join information:
  - **Type the set name** under Column #1/Set.
  - **Type the derivation source number of the set owner** under Of Der.
  - **Enter the operator ST** to indicate a set join.
  - **Leave the Column #2 field blank.**
  - **Type the derivation source number of the set member** under Of Der.

The sample Extended Table Derivation screen shown below illustrates the information entered to join the EMPLOYEE and DEPARTMENT tables on the DEPT-EMPLOYEE set.

XDER CA - Automatic System Facility		CA		
DC560802 ADD/MODIFY DERIVATIONS		*Extended Table Derivation*		
Table Name: EMPLOYEES AND DEPARTMENTS				
		_ PF1 - Help	Page: 1	of 1
Source Name.: EMPLOYEE		Derivation Source 01	1	
Source Owner:				
Subschema...: EMPSS01	Schema: IDMSR	Version: 1		
Source Name.: DEPARTMENT		Derivation Source 02		
Source Owner:				
Subschema...: EMPSS01	Schema: IDMSR	Version: 1		
	Column #1/Set	Of Der	Oper	Column #2
r1	DEPT-EMPLOYEE2	023	ST4	Of De
5				0

- 1 Identifying the source tables

Joining the tables on an owner-member set relationship

- 2 Set name
- 3 Number of set owner
- 4 Join operator
- 5 Number of set member

#### Considerations

1. For ASF to find the non-SQL defined records, you must:
  - Include the ASF dictionary segment in the DBNAME that holds the records.
  - Issue a DCUF SET DBNAME command before entering ASF to set the session default database to the one that holds the records.
2. Deriving from non-SQL defined records requires a knowledge of database navigation. The order in which you specify source record and join information affects the data that is returned. The following considerations apply when deriving from non-SQL defined records:
  - The first record identified is obtained through an area sweep. All other records must be obtainable through set relationships so that one logical retrieval path can be built. If ASF finds that it will perform two area sweeps to obtain source records, then an error is issued for the join information because all the records are not connected in one logical path.
  - Since the first record is obtained through an area sweep, the order in which the records are identified determines how the data will be obtained.
  - The order in which the set joins are specified must coincide with the order in which the source records are identified.
  - A set join is disallowed if the owner and member derivation sources are already directly or indirectly connected. Before accepting a set join as valid, ASF analyzes previously defined set joins to determine if the owner or member derivation records are already directly or indirectly connected. If the records are already connected, the join is not allowed.

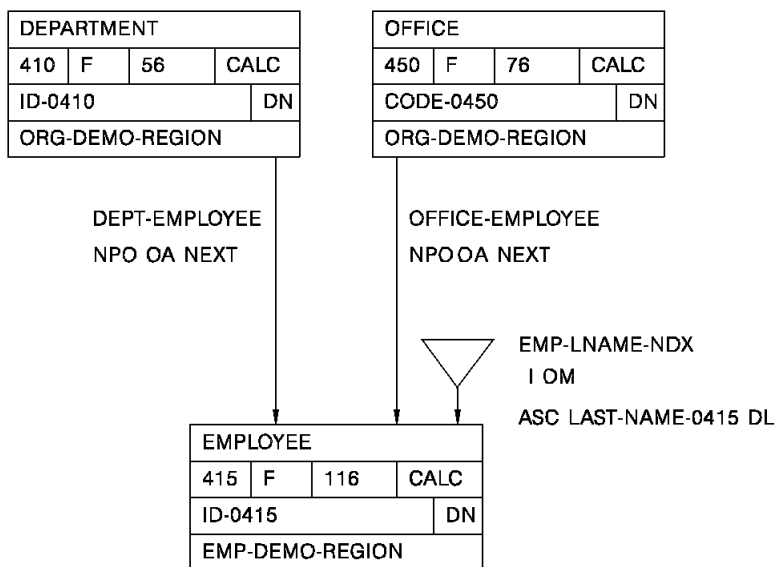
The following examples illustrate the above.



## Examples

### Identifying Source Records in Different Orders to Change Results

The following diagram illustrates the records being used to derive a view:



The records are identified in the following order:

Source Record	Derivation Source #
DEPARTMENT	01
EMPLOYEE	02
OFFICE	03

The records are joined on the following set relationships:

Column #1/Set	Of Der	Oper	Column #2	Of Der
DEPT-EMPLOYEE	01	ST		02
OFFICE-EMPLOYEE	03	ST		02

The records are obtained as follows:

OBTAIN EACH DEPARTMENT WITHIN *area-name*  
 OBTAIN EACH EMPLOYEE WITHIN DEPT-EMPLOYEE  
 OBTAIN OWNER WITHIN OFFICE-EMPLOYEE

In this example, any employee not connected to a department will not be returned.

The records are identified in the following order:

Source Record	Derivation Source #
EMPLOYEE	01
OFFICE	02
DEPARTMENT	03

The records are joined on the following set relationships:

Column #1/Set	Of Der	Oper	Column #2	Of Der
OFFICE-EMPLOYEE	02	ST		01
DEPT-EMPLOYEE	03	ST		01

The records are obtained as follows:

```
OBTAIN EACH EMPLOYEE WITHIN area-name
OBTAIN OWNER WITHIN OFFICE-EMPLOYEE
OBTAIN OWNER WITHIN DEPT-EMPLOYEE
```

In this example, all employees are returned, though not every employee may be connected to a department or an office.

### Defining an Ambiguous Path

The previous diagram illustrates the records being used for derivation.

The records are identified in the following order:

Source Record	Derivation Source #
DEPARTMENT	01
OFFICE	02
EMPLOYEE	03

The records are joined on the following set relationships:

Column #1/Set	Of Der	Oper	Column #2	Of Der
DEPT-EMPLOYEE	01	ST		03
OFFICE-EMPLOYEE	02	ST		03

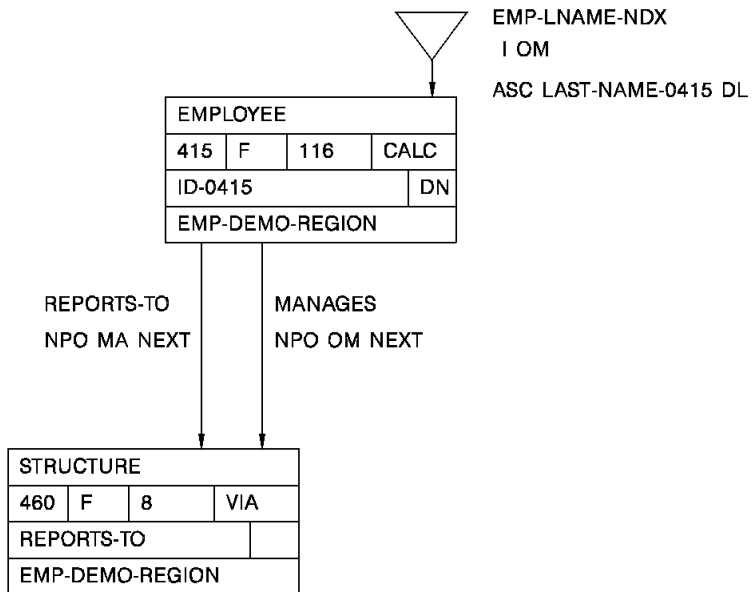
ASF would attempt to build a retrieval path as follows:

```
OBTAIN EACH DEPARTMENT WITHIN area-name
OBTAIN EACH OFFICE WITHIN area-name
OBTAIN EACH EMPLOYEE WITHIN DEPT-EMPLOYEE
```

The OFFICE-EMPLOYEE set join would cause an error because ASF is unable to connect the OFFICE record, which was already obtained, to the EMPLOYEE record obtained through the DEPT-EMPLOYEE set. Since the DEPARTMENT and OFFICE records are not connected through any set relationship, a logical path cannot be built.

**Joining Records that Are Already Directly Connected**

The following diagram illustrates the records being used for derivation:



The records are identified as follows:

Source Record	Derivation Source #
EMPLOYEE	01
STRUCTURE	02

The join information is entered as follows:

Column #1/Set	Of Der	Oper	Column #2	Of Der
REPORTS-TO	01	ST		02
MANAGES	01	ST		02

The MANAGES set join would not be allowed because the EMPLOYEE and STRUCTURE records are already directly connected through the first join defined.

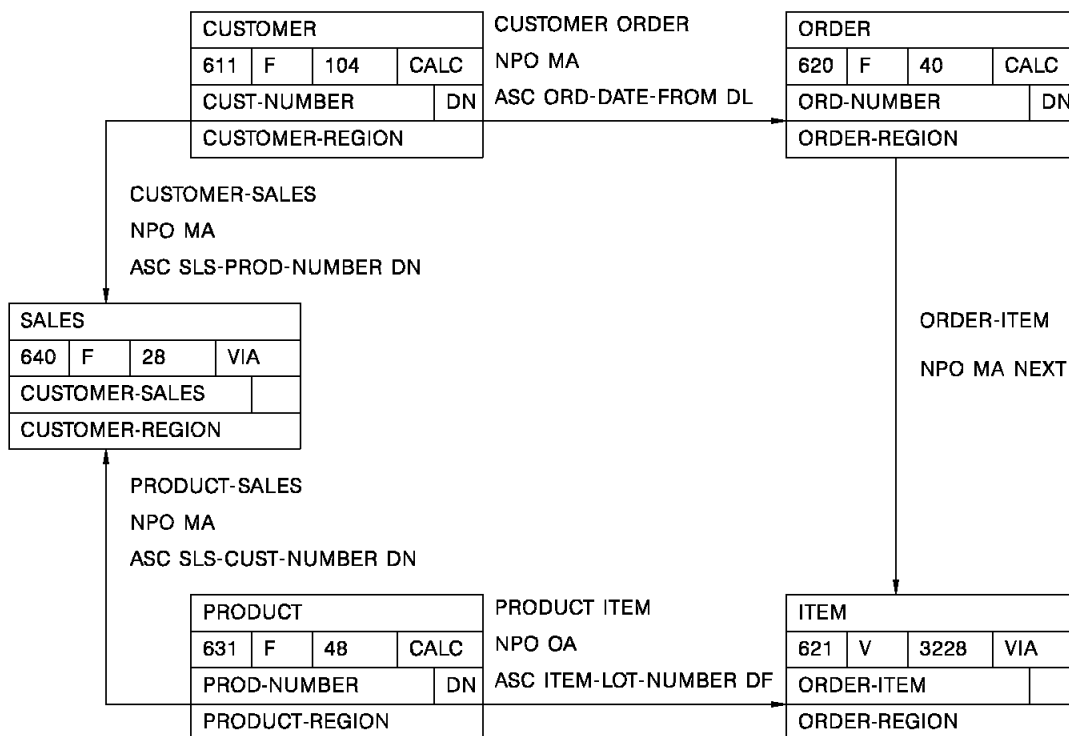
**Note:** To include the MANAGES set in the table, identify the EMPLOYEE record again as a separate derivation source (derivation source #3) and specify the following join information:

Column #1/Set	Of Der	Oper	Column #2	Of Der
REPORTS-TO	01	ST		02
MANAGES	03	ST		02

Note that the resulting view returns two records of the same record type. Only the last record obtained will be updated in the database when updates are made through the view.

**Joining Records that Are Already Indirectly Connected**

The following diagram illustrates the records being used for derivation:



The records are identified as follows:

Source Record	Derivation Source #
CUSTOMER	01
SALES	02
PRODUCT	03
ITEM	04
ORDER	05

The join information is entered as follows:

Column #1/Set	Of Der	Oper	Column #2	Of Der
CUSTOMER-SALES	01	ST		02
PRODUCT-SALES	03	ST		02
PRODUCT-ITEM	03	ST		04
ORDER-ITEM	05	ST		04
CUSTOMER-ORDER	01	ST		05

The CUSTOMER-ORDER set join would not be allowed because the CUSTOMER and ORDER records are already indirectly joined through the previously defined set relationships.

When you derive a view from non-SQL defined records, the elements of the schema records become the columns for the view. The following considerations apply:

- **Occurring fields**, including nested occurring fields, are defined as separate columns. For example, a record element is defined as follows:

```
02 REGION
   OCCURS 3.
   04 REGIONAL-MANAGER PIC X(20).
   04 SALES-MANAGER    PIC X(20)
   OCCURS 2.
```

The above element would result in the following columns:

```
REGION.1
REGIONAL-MANAGER.1
SALES-MANAGER.1.1
SALES-MANAGER.1.2
REGION.2
REGIONAL-MANAGER.2
```

SALES-MANAGER. 2. 1  
SALES-MANAGER. 2. 2  
REGION. 3  
REGIONAL -MANAGER. 3  
SALES-MANAGER. 3. 1  
SALES-MANAGER. 3. 2

**Note:** Varying OCCURS clauses (for example, OCCURS DEPENDING ON) are not supported by ASF.

Multiple levels of OCCURS clauses, such as the SALES-MANAGER columns above, are supported by ASF. Online mapping, which creates the screen format for the ASF-generated application, supports first-level OCCURS clauses only. The SALES-MANAGER columns are part of the table definition, but cannot be displayed online through the ASF-generated application.

- You cannot delete **group fields**. You can remove a group field from the screen display in the application generated by ASF by erasing its display sequence number.
- ASF supplies the **external picture** for a column, which is displayed in the Display Format field on the Extended Column Definition screen. ASF does not use the external picture defined in the record for the element from which the column is derived.
- The **external characteristics** of an element, such as the element name, can be changed when defining a column, but the internal Stored Column Name and Stored Format, defined on the Extended Column Definition screen, cannot be changed.
- Fields defined as **bit** are ignored by ASF and are not included in the view definition.
- Columns can be deleted from the view definition, but the fields remain part of the subschema built for the view.

**Note:** Views derived from non-SQL defined records cannot be used as source tables for other views.

## Creating Indexes

### Advantages

The **Extended Key Definition** screen is used to index a table. Place indexes on a table:

- To have control over the data being stored in a table. By placing a **unique key** on a table, you prevent duplicate rows from being included in the table.
- To sort the data stored in a table. The rows can be sorted according to the value of a key column, in either ascending or descending order.
- To help ASF process join and selection criteria efficiently.

### Types of Indexes

On the Extended Key Definition screen, you can define the following index keys:

- **Multiple keys** for a table
- **Unique or nonunique keys** for a table
- **Concatenated keys** that consist of multiple columns

You can also specify the order in which you want the rows of a table to be stored and displayed.

Keys can be defined for stored tables only. Since data is not actually stored in a view, keys are unnecessary and are not allowed for views. Columns that are defined as work fields cannot be included in an index key.

A sample Extended Key Definition screen is shown below.

XKEY CA - Automatic System Facility		CA					
DC560906 ADD/MODIFY KEYS		*Extended Key Definition*					
Table Name: JOB		PF1 - Help Page: 1 of 1					
Storage Sequence: 00001		Next/Prior/First/Last/Key Number					
Display Sequence: 00001		Same/Key Number					
		Additional Keys					
		Key Number: 01 02 03 04 05 06					
		Fast Duplicates: NO LAS LAS LAS LAS LAS					
		Access Ascending.: YES YES YES YES YES YES					
Column Name	Keys	Column Order					
JOB NUMBER	-	11					
DEPARTMENT	-	22					
TITLE	x3						
SALARY	-	1					
CURRENT STATUS (FILLED OR OPEN)	x						

1. A unique key
2. A concatenated key consisting of two columns
3. Individual non-unique keys

The Extended Key Definition screen is used to specify the following:

- Fast access keys
- Additional index keys
- Storage and display sequence for data

Each of these topics is discussed separately below.

## Fast Access Keys

### Use for Join, Select Columns

Columns that are used for joining and selecting data should be defined as **fast access keys**. A fast access key consists of one column, and is indexed as duplicates last, sorted in ascending order.

You can define more than one fast access key for a single table. Type **x** in the Fast Access Keys field for each column you want to define as a fast access key.

## Additional Index Keys

The Additional Keys portion of the Extended Key Definition screen gives you more flexibility in your key definitions. You can define additional keys with the following options:

- Duplicates last, first, or not allowed
- Index sort order either ascending or descending

### Concatenated Keys

Additional keys can consist of one column or more than one column. Keys that consist of more than one column are **concatenated keys**. When defining a concatenated key, you specify the order in which the columns are to be combined, as shown below:

NAME	2
ADDRESS	
CITY/ST/ZIP	
TELEPHONE	1

Each additional key is identified by the **key number** under which it is defined. The key number can be used to specify the storage sequence and display sequence of the table.

### Defining Additional Keys

To define an additional key:

- **In the Duplicates field**, accept the default value LAS (duplicates last), or enter FIR (first) to store duplicates first or NO to specify that duplicates are not allowed.
- **In the Ascending field**, accept the default value YES to have the order ascending, or enter NO to make the order descending.
- **Under the Column Order heading**, number the columns that make up the key in the order that you want them concatenated. If the key consists of only one column, enter a field order of 1 for that column.



### Unique Keys are Additional Keys

When a unique key is specified on the Column Definition screen, it displays as an additional key on the Extended Key Definition screen. If the unique key consists of more than one column, the Column Order is the order in which the columns are defined. You can modify the definition of the unique key on the Extended Key Definition screen if desired.

## Defining the Storage and Display Sequence

The **storage sequence** indicates the order in which data will be stored in the database. The storage sequence of a table can be any of the following:

- **Next** stores a new row after the most recently displayed row.
- **Prior** stores a new row before the most recently displayed row.
- **First** stores a new row before the rows already stored.
- **Last** stores a new row after the rows already stored.
- **Sorted** stores rows according to a specified key.

You can specify the storage sequence by entering **next, prior, first, last** or the key number of a specified key in the Storage Sequence field. The storage sequence defaults as follows:

- If no key is defined for the table, the storage sequence is NEXT.
- If a unique key is defined on the Column Definition screen, the storage sequence is according to that key, which displays as the first additional key.
- If additional keys are defined for the table, the storage sequence is according to the key you specify, or it defaults to NEXT.

### Display Sequence

The display sequence is the order in which data will be displayed through the online application created by ASF for the table. The display sequence can be either of the following:

- The **same as the storage sequence** to display rows in the order in which they are stored
- The **key number of a defined key** to display rows in a particular sorted order

The display sequence defaults to SAME, but you can specify a different display sequence by entering a key number other than that specified for the storage sequence.

**Note:** A table has only one storage sequence and one display sequence. You cannot specify the display sequence for a table at run time, although it is influenced by run-time selection criteria.

# Chapter 10: Redefining, Regenerating, and Deleting Tables

---

ASF enables users to create and manage data tables. Managing involves redefining, regenerating, and deleting tables. Through ASF, users can tailor existing tables to meet changing business needs. This section covers the following topics:

- Redefining tables
- Regenerating tables
- Deleting tables
- Modifying and deleting source tables

This section contains the following topics:

[Redefining Tables](#) (see page 155)

[Regenerating Tables](#) (see page 158)

[Deleting Tables](#) (see page 160)

[Modifying and Deleting Source Tables](#) (see page 161)

## Redefining Tables

During the definition process, you can adjust and readjust the table definition until it fits your needs exactly. Once the table is generated, you can **redefine** the table as your needs change.

You can change the definition of a generated table in several ways.

### Table Definition

On the Table Definition screen (TDEF), you can change only the derivation information (sources, join fields, and join criteria); column definitions are automatically modified to reflect changes in the derivation information. You cannot change the table type; for example, you cannot redefine a view as a stored table.

### Extended Table Definition

On the Extended Table Definition screen (XTAB), you can modify:

- Table access restrictions (display, load, change, and erase access)
- Generate map/dialog instruction

An ASF administrator can change the area in which the table is stored.

### Column Definition

On the Column Definition screen (CDEF), you can modify:

- Column name
- Column width (stored tables only)
- Display sequence number
- Unique key specification

When you modify columns, ASF updates the table definition upon regeneration. ASF performs data conversion for updates as shown below:

Update Command	Column Criteria	Keys	Fields *	Selection
ADD	X	X		X
DELETE	X	X		X
MODIFY		X	X	X

\* Length and data type

You can add and delete unique keys. If the unique key is a concatenated key, you should delete the entire key, not part of it. If you add a unique key, and a duplicate row is encountered, the regeneration of the table is interrupted and the duplicate row is written to the log.

### Extended Column Definition

On the Extended Column Definition screen (XCOL), you can change the following information for stored tables:

- Display name and format
- Stored name and format
- Initial value and/or calculation
- Edit and code table associations
- Mapping attributes

For views, you can change the following information:

- Display name and format
- Calculation
- Edit and code table associations
- Mapping attributes

### **Extended Table Derivation**

On the Extended Table Derivation screen (XDER), you can modify:

- Source tables/records
- Join criteria

Note that changes to the derivation information are reflected automatically in the column definitions.

### **Extended Selection Definition**

On the Extended Selection Definition screen (XSEL), you can define new selection criteria for the table, or change or delete selection conditions that have been previously defined.

### **Extended Key Definition**

On the Extended Key Definition screen (XKEY), you can change the storage and display sequence of the table, define new index keys with duplicates allowed, or modify or delete existing keys.

### **Row-level Security**

On the Row-Level Security screen (RSEC), you can change the security name and generic column names assigned to the table.

### **Redefinition Status**

Modifying the definition of a generated table places the table in **REDEFINITION** status. You can continue to use the table through the Load Data and Display/Change Data functions of ASF, but the table will not reflect the changes you have made. The Select Data and Query functions are not available for tables in REDEFINITION status.

You can continue to access the table outside ASF through the generated subschema, but this is the previously generated subschema and it does not reflect the changes you have made to the table definition.

## Regenerating Tables

### Effecting Changes

After you modify the table definition, you must regenerate the table before the changes will take effect. To regenerate the table, select the Generate function on the Table Definition screen. Regenerating a table involves some or all of:

- Updating the record and element definitions in the data dictionary
- Modifying the schema
- Regenerating the subschema
- Regenerating the map and dialog
- Restructuring the data in the database

### Regeneration Can be Expensive

Because regeneration potentially involves extensive processing on ASF's part, use some constraint in the frequency with which you redefine and regenerate data tables.

Any action causing a data conversion during table regeneration causes the data to be restructured. This can take a considerable amount of time, particularly for tables with a large number of rows. Since restructuring the data involves copying the entire table, the amount of space required should also be considered when regenerating tables with large amounts of data.

### Regenerate with No Restructure

You can choose to regenerate the table definition *without* restructuring the table data. To obtain this definition-only regeneration, follow these steps:

1. Make sure the Process Standard Area Data field on the Extended Table Definition screen is set to NO. This causes ASF to modify only the table definition, when the Generate function is invoked. Note that you must have administrative authority to access this field.
2. Select the Generate function from the Table Definition screen. ASF will now regenerate the table definition and return a status of GENERATED.
3. Prepare the area for new data as follows:
  - If there is only one table in the area, use the formatting utility (FORMAT) to reinitialize the area. FORMAT is documented in *CA IDMS Utilities Guide*.
  - If there is more than one table in the area, run MAINTAIN ASF to erase the table data.

### Reloading Data

Now, you can optionally reload the table data. The steps you follow to reload the table depend on the facilities you choose to use:

- **To reload data by using MAINTAIN ASF:**
  1. Run MAINTAIN ASF to reload the data, as documented in the chapter "MAINTAIN ASF".
  2. Invoke the ASFOAKD dialog to update the table definition with the new location of the table data. You must supply the appropriate table definition number when you invoke this dialog.
- **To reload data by using ASF screens, CA Culprit, CA OLQ or supported PC software :**
  1. Invoke the ASFOAKD dialog to update the table definition with the new location of the data table.
  2. Reload the table with the appropriate facility.

## Unsuccessful Regenerate

### Possible Reasons

If you receive the message ERROR ON GENERATE, your table did not generate properly. This could be due to several reasons:

- Creating a duplicate key by removing a field from a unique concatenated key
- Adding a unique key
- Lessening the width of a numeric field causing a truncation error when moving the numeric value to its new field

### Use Log

At this point, ASF has interrupted the data conversion to write the rows in question to the log. You can look at this log to determine where the problem lies. When the ERROR ON REGENERATE warning displays:

1. Look at the Message screen ([PF6]). The Message screen displays a WARNING ON REGENERATE statement. This means that some of the rows you tried to add were not added to the table.
2. Press [Clear] to return to the Table Definition screen.
3. Finish generating the table ([PF3]) to continue processing where it was interrupted.
4. After the table is generated, look at the log to determine the rows that need to be reentered. (Duplicate rows are written to the log, and cannot be added to the table until they have been corrected.)
5. Reenter the rows, taking care to correct the problems.

## Deleting Tables

### Delete Definition and Data

To delete a table, select the Delete Table Definition function from the Table Definition screen. The Delete Table Definition function does all of the following:

- Removes the generated map, dialog, and subschema
- Deletes the table definition from the schema, if the table is a stored table
- If the table is stored, erases all the data stored in the table from the database
- Removes the catalog entry for the table

### Delete Definition Only

You can choose to delete the table definition *without* erasing the table data. To perform a definition-only delete, follow these steps:

1. Make sure the Process Standard Area field in the Extended Table Definition screen is set to NO.
2. Select the Delete Table Definition function from the Table Definition screen.

Once you have deleted the table definition, you can delete the table data as follows:

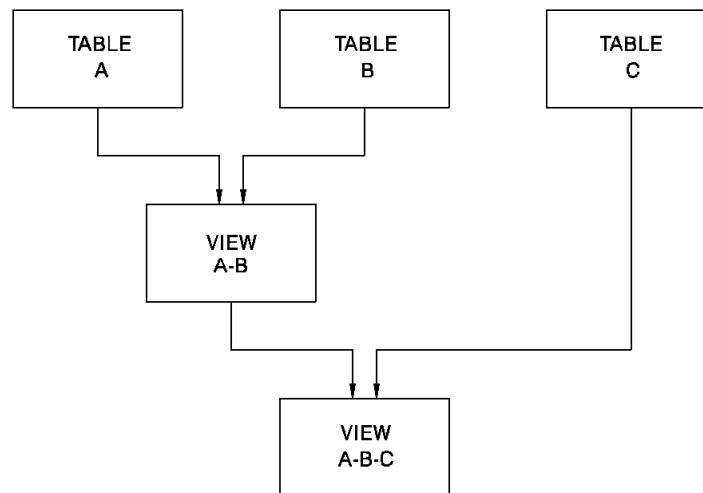
- If there is only one table in the area, run `FORMAT` to reinitialize the standard area.
- If there is more than one table in the area, use `MAINTAIN ASF` to erase the table data.



## Modifying and Deleting Source Tables

### May Affect Views

A source table is a table from which another table is derived. Any table, whether a stored table or a view (except views derived from non-SQL defined records), can be used as a source from which to derive another table. Since views can use other views as source tables, the original source table may be logically removed from a view, as illustrated below.



### Regenerate Affected Views

Modifying or deleting a table, therefore, can have far-reaching effects. Every view that uses the modified or deleted table as a source, either directly or through another view, is given a **MUST REGENERATE** status. When you try to access a view with a **MUST REGENERATE** status through the Display/Change Data, Select Data, or Query functions, you are notified that the view must be regenerated before it can be used.

### Views Change Automatically

The definition of the view reflects the changes made to the source table, as follows:

- If the source table is modified, the portion of the view definition derived from that source table is modified automatically except when one or more columns are added to the source table. In this case, select the view flagged for regeneration, erase the source table name from the table definition screen (TDEF) or extended table derivation screen (XDER), and then re-enter the same name to update the derivation. When the size of a column in a stored table is modified, the view's column definition record reflects the original display format. Go to the extended column definition screen (XCOL) and modify the display format.

When derivation information is deleted and re-added, it may be necessary to go to the column definition screen (CDEF) and renumber the display sequence numbers before regenerating the view. When the derivation is updated, ASF uses a value from the view's definition record to determine the beginning sequence number for columns. It will now be one greater than the number of columns in the original view. As a result, some or all of the data may not be displayed.

Definition selection criteria and work columns must be redefined after the derivation is re-added.

- If the source table was deleted, all references to the source table in the view definition are deleted automatically.

Since the source table modifications are reflected automatically in the definition of the view, you do not have to make the same changes to both tables. You can select the Generate function on the Table Definition screen and regenerate the view and its supporting structures.

### **Regenerate in Proper Order**

Views derived from views must be regenerated in the order that they were defined. For example, in the figure above, if Table A is modified, both View A-B and View A-B-C would be placed in MUST REGENERATE status. *View A-B must be regenerated before View A-B-C.*

### **View Meaning May Change**

Changes to a source table can affect the meaning of a view. If the source table is modified, the portion of the view definition derived from that source table is modified automatically except when one or more columns are added to the source table. In this case, select the view flagged for regeneration, erase the source table name from the Table Definition screen or Extended Table Derivation screen, and then reenter the same name to update the derivation.

### **May Have to Renumber Columns**

Before regenerating the view, it may be necessary to renumber the display sequence numbers on the Column Definition screen. When the derivation is updated, ASF uses a value from the view's definition record to determine the beginning sequence number for columns. It will now be one greater than the number of columns in the original view. As a result, some or all of the data may not be displayed.

### **May Delete View**

In some cases, you may choose to delete a view as a result of changes to a source table. For example, in the figure above, if Table A is deleted, View A-B-C could be modified so that it is derived directly from Table B, becoming View B-C; View A-B could be deleted since it would probably be the same as Table B.

**Be Careful**

Since one table can be used as a source table in many views, the effects of modifying or deleting a table should be considered before any action is taken.



# Chapter 11: Administrative Functions of ASF

---

## Screens to Use

Administrative functions are performed on the following ASF screens:

- Administrator Control—CNTL
- Catalog Control—CTLG
- User Limits and Defaults—ULIM
- User Maintenance—USER
- System Table Maintenance—SYST

Refer to the chapter "Administrative Screens" for a complete description of each administrative screen.

These screens are restricted to users with administrative authority only. Using these screens, the database administrator and other authorized personnel can:

- Initialize the definition area.
- Specify the default area for the system.
- Acquire and distribute administrative authority.
- Initialize row-level security for the system and for individual users.
- Specify system and user defaults.
- Perform general administrative activities for the ASF system.
- Restrict ASF users.
- Delete all object entities in the catalog.
- Control row-level security.

Each of these activities is discussed separately on the following pages.

This section contains the following topics:

[Initializing the Definition Area](#) (see page 166)

[Specifying the Default Area for the System](#) (see page 167)

[Acquiring and Distributing Administrative Authority](#) (see page 168)

[Specifying System and User Defaults](#) (see page 169)

[Performing General ASF Administration](#) (see page 171)

[Restricting ASF Users](#) (see page 172)

[Deleting All Object Entities in the Catalog](#) (see page 179)

## Initializing the Definition Area

### The IDMSR Schema

A system-supplied schema, IDMSR, is provided with the installation of CA IDMS/DB. This schema describes the areas, sets, and records that ASF uses to define data tables:

- Descriptions of the **definition records** used to store information about data tables defined through ASF
- The **definition area**, IDMSR-AREA, where the definition records are stored
- One **secondary area**, IDMSR-AREA2, where the data tables are stored

Before ASF can be used to define data tables, the definition records must be loaded into the definition area by performing the Initialize Definition Area function on the Administrator Control screen.

In addition to loading the definition records, the initialization function also establishes defaults for the ASF system, which can be changed at any time.

## The Steps of Initialization

### Step 1. Sign on

Sign on to ASF as the user **CULL DBA** (password=DBAPASS).

When ASF is installed, the user ID 'CULL DBA' is provided so that the DBA can access the administrative functions of ASF.

After 'CULL DBA' signs on, the **Administrator Control** screen comes up automatically. Notice that the Definition Area Name field displays IDMSR-AREA, which is the definition area defined in the schema.

### Step 2. Access Administrator Control

Complete the fields on the Administrator Control screen.

#### Default Models

You can specify default models used to generate subschemas, dialogs, and maps for tables (see "Defaults for Models Used by ASF" later in this chapter). If no models are specified by the user, the following defaults are supplied by ASF:

- MODEL MAP.....: ASF@@MAP VERSION: 1
- MODEL DIALOG.....: ASFDGMOD VERSION: 1
- MODEL SUBSCHEMA: ASFSSMOD VERSION: 1

### Step 3. Initialize the Area

Press [PF3] to invoke the Initialize Definition Area function. This function stores the following records in the definition area:

- The OOK owner records for the relational definition records
- One RDEFREC (record definition record) for each definition record type
- The AREAREC (area record)
- The GCTLREC (generation control record)

The definition records need to be loaded into the definition area only. The Initialize Definition Area function of the Administrator Control screen is used only when the ASF system is initialized.

#### **Add Non-SQL Defined Records**

Non-SQL defined records that are to be made available to ASF should be added to the schema IDMSR as soon as the system is initialized, before any tables are created through ASF. This ensures that you will not be adding record, area, and set descriptions whose names duplicate names of schema components created by ASF.

## Specifying the Default Area for the System

Once the definition area has been initialized, you should identify the secondary area, **IDMSR-AREA2**, to the ASF system:

1. Enter IDMSR-AREA2 in the Secondary Area Name field
2. Press [PF2]

This first area you add will automatically be the default area for the ASF system. To change the default area later, overwrite the name of the area in the default area name field.

## Acquiring and Distributing Administrative Authority

### Acquire

Once the ASF system has been initialized and the default area has been specified, as the DBA, you should acquire administrative authority:

- Access the User Maintenance screen (either press [PF7] or type **user** in the upper left corner).
- Enter your user ID in the User Name field (your user ID must be established in the dictionary with authority for ASF).
- Select the Add User function.
- Access the Catalog Control screen by typing **ctlg** in the upper left corner.
- Select Give Administrator Authority and type your user ID In the User Name field.

Once you have acquired administrator authority, you should sign on to ASF as yourself and remove administrator authority from CULL DBA, since the user ID CULL DBA is common knowledge.

Alternatively, you can let CULL DBA keep administrative authority, but change the default password (DBAPASS).

### Distribute

Any user with administrator authority can give and remove that authority from others. The DBA can distribute the administrative responsibilities for ASF by giving authority to other users on the DBA staff. Before users can be given administrator authority, they must:

- Be established in the dictionary with authority for ASF
- or
- Be entered in the catalog either through the Add User function on the User Maintenance screen or by signing on to ASF



## Specifying System and User Defaults

ASF has two levels of defaults, as follows:

- **System defaults**, which are specified on the Administrator Control screen. System defaults are set at initialization but can be changed at any time.
- **User defaults**, which are specified on the User Limits and Defaults screen. User defaults are assigned on an individual user basis and override the defaults set for the system.

The following two types of defaults are involved:

- Defaults for space allocation
- Defaults for models used by ASF

The defaults for models are discussed below.

### Defaults for Models Used by ASF

Default models are provided with the ASF system. The models are used by ASF to generate the subschema, dialog, and map for each table. The models are stored in the data dictionary as the following modules and map:

- **ASFSSMOD** is the dictionary module source that contains the model subschema.
- **ASFDGMOD** is the dictionary module source that contains the model dialog.
- **ASF@@MAP** is the model map.

### Options

ASF needs a model to generate an application. The following options are available for specifying the default models used by ASF:

- **Use the models provided** with the ASF system. These models become the default models used by the system and no other models need to be specified. When the ASF system is initialized, the model names listed above are displayed on the Administrator Control screen.
- **Modify the models provided** to meet the needs of an individual user site, as follows:
  - Change the names of the subschema and dialog to adhere to standard naming conventions.

**Important!** The names of the model subschema and dialog can be changed to conform to the naming conventions of an individual site; the model subschema and dialog should not be modified in any way.
  - Modify the model map. When you modify the model map, it is recommended that you store the modified model map as a separate version, or under a different name, and keep the original model map intact.

You can designate the modified model as the default model for the system by entering the appropriate name and version number on the Administrator Control screen.

- **Store several models** by storing the model subschema and dialog under several different names or by defining several different model maps. You can specify, on the Administrator Control screen, the models to be used as the system defaults. You can assign different models as the defaults for individual users by using the User Limits and Defaults screen.

**Note:** Source for the model map, dialog, and subschema must be in the data dictionary. The dialog and subschema must be stored as modules, qualified by LANGUAGE IS ASF

### The Model Map(s)

Typically, you will want to modify the model map to make it site-specific. The default model map provided with ASF, **ASF@@MAP**, is shown below.

```
                                CA
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

Since the source definition of the model map is stored in the data dictionary, the model map can be modified through the CA IDMS/DC and CA IDMS UCF online mapping facility to meet the standards of the user site. For example, Acme Insurance modified the model map as shown below.

```

ACME INSURANCE
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

```

**Model Map Considerations**

The following should be considered when modifying the model map:

- The at signs (&at sign.) are converted to the table name when the individual screen is formatted. If they are deleted, the table name is not displayed on the Data Display screen.
- ASF begins formatting the screen for the individual table on the first available blank line at the end of the model map. The heading information, message text, and table name should be defined in as few lines as possible to allow the maximum number of lines for displaying the column names and response fields for the table.

## Performing General ASF Administration

The following functions are available to help the DBA maintain the ASF system:

- Adding secondary areas
- Unlocking table definitions

### Adding Secondary Areas

The definition area contains occurrences of the definition tables needed to define data tables. Data tables are stored in **secondary areas**. One secondary area, IDMSR-AREA2, is provided when ASF is installed. It is recommended that IDMSR-AREA2 be used as the default area for the system. You can also add additional secondary areas for use by ASF.

To add a secondary area, follow the steps outlined in the *CA IDMS Database Administration Guide*. When the area has been initialized and added to the IDMSR schema, identify the area to ASF by completing the Secondary Area Name field on the Administrator Control screen and selecting the Add Secondary Area function ([PF2]).

## Unlocking Table Definitions

When a user modifies a table definition through ASF, a lock is placed on the table for that user ID. The lock is removed automatically when the user selects another table or ends the ASF session. If the ASF session is interrupted while a lock is in place, the table definition remains locked for update until the user completes the modification and unlocks the table, or until the DBA unlocks the table through the Administrator Control screen.

To unlock a table definition, the DBA takes the following steps:

1. Enters the definition number assigned to the locked table in the Unlock Table Definition Number field on the Administrator Control screen
2. Selects the Unlock Table Definition function by pressing [PF4]

The lock on the table definition is released and can be updated by another user.

## Restricting ASF Users

The following functions are available to enable the DBA to restrict individual users in their use of ASF:

- Limiting passkey authorization
- Limiting row-level security authorization
- Locking and unlocking users
- Deleting users from the catalog

## Limiting Passkey Authorization

When a user is defined in IDD as having authority for ASF, the user automatically has full passkey authorization. An ASF administrator can limit what a user can do through ASF by disallowing some of the user's global passkeys. The restrictions apply throughout the ASF system, not only to the user's own tables, but also to tables owned by others. The passkey authorization a user has for ASF determines the passkeys the user can be granted by other users.

For example, if the DBA wanted to prevent you from defining tables through ASF, the administrator would disallow the global CREATE passkey for your user ID. Without a CREATE passkey:

- You cannot define tables for yourself.
- You cannot be given a CREATE passkey to define tables for another user.

However, you can own tables that have been defined for you, and you can also be given access to tables owned by others.

#### How To Do It

To limit a user's passkey authorization, do the following:

1. Access the User Limits and Defaults screen
2. Enter the user's ID in the User Name field
3. Type **n** next to the passkey(s) being disallowed

## Limiting Row-level Security Authorization

A database administrator must establish row-level security on the system before users can begin assigning row-level security to their tables. Establishing row-level security involves two steps:

- Generating system administration tables
- Authorizing users to employ row-level security

Row-level security is managed through the three system tables:

- SRT—\$SECURITY-RUNTIME-TABLE\$
- OST—\$OBJECT-SECURITY-NAME-TABLE\$
- SXRF—\$SRT-OST-CROSS-REFERENCE\$

## Generating System Administration Tables

#### Finding the Tables

You can access the system administrator tables through the System Table Maintenance (SYST) screen. To access the SYST screen:

1. Sign on with the user ID **cull dba** and the password assigned to CULL DBA. Your system is installed with the password DBAPASS, but this is often changed for security reasons.

- Enter **syst** in the screen name field and press [Enter]. The SYST screen is shown below:

```

SYST  CA - Automatic System Facility          CA
DC560727 SELECT A SYSTEM TABLE             *System Table Maintenance*

Table Name:

  _ PF1 - Help                               _ PF4 - Display/Change Data
  _ PF2 - Define Table                       _ PF5 - Select Data
  _ PF3 - Load Data                         _ PF6 - Query

  _ $UNSTRUCTURED-IDB-OBJECT$
  _ $SECURITY-RUNTIME-TABLE$
  _ $OBJECT-SECURITY-NAME-TABLE$
  _ $SRT-OST-CROSS-REFERENCE$
    
```

From the SYST screen you can access table definition screens for the system tables.

### Security Runtime Table

To generate the \$SECURITY-RUNTIME-TABLE\$ (SRT):

- Choose the SRT from the SYST screen and press [PF2] to access the Table Definition (TDEF) screen:

```

TDEF  CA - Automatic System Facility          CA
DC560029 INQUIRY ONLY, SYSTEM TABLES CAN ONLY BE GENERATED THROUGH THIS SCREEN
                                           ** Table Definition **

  _ PF1 - Help                               _ PF4 - Extended Table Definition
  _ PF2 - Define Columns                     _ PF5 - Delete Table Definition
  _ PF3 - Generate                           _ PF6 - Message Screen

Table Name.: $SECURITY-RUNTIME-TABLE$
Table Owner: IDBSYSTEM                      Defn Number: 20
View/Stored: STORED                          Status.....: UNGENERATED
Comments...:

                                           Table Derivation

Source Table #1
Table Name.:
Table Owner:

Source Table #2
Table Name.:
Table Owner:

Column #1:
Column #2                                     Where Column #1 EQ Column #2
    
```

- Press [PF3] to generate the table.
- Press [Clear] to return to the SYST screen.

### Object Security Table

To generate the \$OBJECT-SECURITY-NAME-TABLE\$ (OST):

- Choose the OST from the SYST screen and press [PF2] to access the Table Definition (TDEF) screen:

```

CA
TDEF CA - Automatic System Facility          ** Table Definition **
DC560029 INQUIRY ONLY, SYSTEM TABLES CAN ONLY BE GENERATED THROUGH THIS SCREEN

      _ PF1 - Help                          _ PF4 - Extended Table Definition
      _ PF2 - Define Columns                 _ PF5 - Delete Table Definition
      _ PF3 - Generate                       _ PF6 - Message Screen

Table Name.: $OBJECT-SECURITY-NAME-TABLE$
Table Owner: IDBSYSTEM                      Defn Number: 21
View/Stored: STORED                         Status.....: UNGENERATED
Comments...:

                                Table Derivation

Source Table #1
Table Name.:
Table Owner:

Source Table #2
Table Name.:
Table Owner:

Column #1:
Column #2:                                Where Column #1 EQ Column #2

```

- Press [PF3] to generate the table.
- Press [Clear] to return to the SYST screen.

**Cross References**

To generate the \$SRT-OST-CROSS-REFERENCE\$ (SXRF):

- Choose the SXRF from the SYST screen and press [PF2] to access the Table Definition (TDEF) screen:

```

CA
TDEF CA - Automatic System Facility          ** Table Definition **
DC560029 INQUIRY ONLY, SYSTEM TABLES CAN ONLY BE GENERATED THROUGH THIS SCREEN

      _ PF1 - Help                          _ PF4 - Extended Table Definition
      _ PF2 - Define Columns                 _ PF5 - Delete Table Definition
      _ PF3 - Generate                       _ PF6 - Message Screen

Table Name.: $SRT-OST-CROSS-REFERENCE$
Table Owner: IDBSYSTEM                      Defn Number: 22
View/Stored: VIEW                           Status.....: UNGENERATED
Comments...:

                                Table Derivation
Source Table #1
Table Name.: $OBJECT-SECURITY-NAME-TABLE$
Table Owner: IDBSYSTEM

Source Table #2
Table Name.: $SECURITY-RUNTIME-TABLE$
Table Owner: IDBSYSTEM

Column #1: OWNER-FROM-OST
Column #2: OWNER                               Where Column #1 EQ Column #2
    
```

- Press [PF3] to generate the table.
- Press [Clear] to return to the SYST screen.

**Authorizing Users for Row-level Security**

**Give Access to SRT**

At installation time, the only user authorized to use row-level security is CULL DBA. If users are to apply row-level security to their tables, they must have the authority to access the SRT. However, the SRT itself is protected by row-level security. As a result, users must be associated with the SRT's security name, SRTSECNM.

To give users authority to access the SRT, the DBA must first access the RLOD screen:

1. Type **rlod** in the screen name field of any screen or choose the SRT on the SYST screen and press [PF3].
2. At the RLOD screen, enter the information listed below. If you are granting access to an individual user, enter the user ID and leave the group ID blank. If you are granting access to a group (you must have CA-ICMS installed), leave the user ID blank and enter the group ID.



- **GROUP**—The group to have access to row-level security.
- **USER**—The user to have access to row-level security.
- **OWNER**—The owner of the SRT table; type **idbssystem**.
- **SECURITY-NAME**—The security name associated with the SRT; type **srtsecnm**.
- **WHERE-CLAUSE**—The user or group ID of the person or group being authorized to establish security. You should normally allow each user to see only the information associated with his own tables. To do this, specify **owner eq user-id**.

If you leave this field blank, you automatically grant DBA authority to that user or group.

3. Press [PF1] to add the row to the SRT table.
4. Press [PF4] to validate the entry.

NO ERRORS ON VALIDATION	CA
\$SECURITY-RUNTIME-TABLE\$	
GROUP	
USER	
OWNER	
SECURITY-NAME	
STATUS	
WHERE-CLAUSE	

**Note:** When a table is deleted, the SRT entries are *not* automatically deleted.

If you delete the CULL DBA entry in the SRT without having previously added another user to the SRT (with administrator authority), you will have to initialize ASF in order to access the SRT again.

## Handling Errors

If any of the catalog entries (group, user, owner) that you entered do not exist or the format of the WHERE clause is incorrect, the system displays the message: ERROR ON VALIDATION, PROCEED TO MESSAGE SCREEN. If this happens, identify and correct the error by performing these steps:

- Press [PF5] to view the error messages on the Message screen. For example, if you tried to enter a user who was not entered in the catalog, the message DC432065 (E) USER xxx, NOT DEFINED IN CATALOG is displayed.
- Press [Clear] to return to the RLOD screen. At the RLOD screen, press [Clear] again to return to the ASEL screen.
- At the ASEL screen, type **rsel** in the screen name field.
- On the RSEL screen, next to the STATUS column, enter **ne 'v'** and press [Enter] twice to display those SRT entries with validation errors.
- Enter the correct information on the RLOD screen. Press [PF2] to change the row and [PF4] to validate.
- Validate the corrected entries by performing the validation process once.

## Locking and Unlocking Users

The DBA may need to temporarily restrict a user's ability to sign on to ASF. To lock a user out of ASF, take the following steps from the Catalog Control screen:

1. Select the Lock User activity from the menu
2. Complete the User Name field

The user is prevented from signing on to ASF until the DBA removes the restriction through the Unlock User function.

## Deleting Users from the Catalog

The DBA may need to permanently restrict a user's access to ASF. To revoke a user's access to ASF, the DBA must take the following steps:

1. **Delete all tables owned by the user.** The user cannot be deleted from the catalog as long as the user owns tables. Alternatively, if CA-ICMS is installed, you can move the tables to a new owner.
2. **Access the User Maintenance screen,** enter the user's ID in the User Name field, and select the Delete User function.

The user no longer has access to ASF and has been removed from the catalog. Note that if the user is *deleted* from IDD, the user's catalog entry is deleted automatically and it is not necessary to perform the Delete User function on the User Maintenance screen.

## Deleting All Object Entities in the Catalog

The Delete All Object Entities In The Catalog function on the Catalog Control screen is *performed only when the definition area has been reinitialized*. For more information on this function and when it can be used, see the *CA IDMS Database Administration Guide*.



# Chapter 12: Accessing Tables Outside ASF

---

The tables created through ASF can be accessed through the following facilities and products:

- The Information Center Management System (CA-ICMS)
- OnLine Query (CA OLQ)
- CA Culprit
- CA IDMS/DB Data Manipulation Language (DML)
- Application Development System (CA ADS)
- Supported PC software

When using a table through these facilities, identify the table as follows:

- In CA-ICMS, CA OLQ, and CA Culprit, provide the **table name** as displayed in the directory.
- In DML and CA ADS, provide the **logical record name**, as displayed on the Extended Table Definition screen.

To access the map, dialog, or subschema generated by ASF for a table, use the following format:

Map	RMnnnnnn
Dialog	RDnnnnnn
Subschema	RUnnnnnn

*Nnnnnn* is the table definition number assigned by ASF on the Table Definition screen.

These names are also available on the Extended Table Definition screen.

This section contains the following topics:

[Information Center Management System](#) (see page 182)

[CA OLQ](#) (see page 182)

[CA Culprit](#) (see page 183)

[CA IDMS/DB Data Manipulation Language \(DML\)](#) (see page 183)

[Application Development System \(CA ADS\)](#) (see page 184)

## Information Center Management System

CA-ICMS links personal computers and minicomputers to the mainframe. ASF is a component of CA-ICMS. When CA-ICMS is installed with CA IDMS/DB, ASF is a component of both the information center and development center systems. Since both systems share the **catalog**, which defines the relationships between tables and users, ASF users can take advantage of full catalog functionality. Full catalog services are provided by the CA-ICMS Manager, a component of CA-ICMS. ASF users can place tables into folders and users into groups.

Because the information center and development center share the same catalog, tables defined through ASF are available to PC users through supported PC software.

## CA OLQ

### Display Tables

OnLine Query (CA OLQ) is designed to retrieve information from the database. With CA OLQ, tables defined through ASF can be displayed in row and column format. CA OLQ also enables you to specify criteria for selecting the rows you want displayed in the table.

### CA OLQ Menu Mode

CA OLQ menu mode is directly accessible from ASF. Access CA OLQ by selecting the Query function from the Activity Selection screen or by entering the screen name OLQ.

### Create, Update Tables

CA OLQ can also be used to create data tables. The definitions of tables created through CA OLQ can be displayed and modified through ASF. Optionally, ASF can generate an application for a CA OLQ-created table.

---

## CA Culprit

### Print Tables

CA Culprit is the CA IDMS report generator. ASF-defined tables can be retrieved by CA Culprit and printed on user-defined reports. CA Culprit enables you to produce a printed copy of the contents of a table.

### Create, Update Tables

CA Culprit can also be used to create and update data tables. The definitions of tables created through CA Culprit can be displayed and modified through ASF. Optionally, you can generate an application through ASF for a table created through CA Culprit.

## CA IDMS/DB Data Manipulation Language (DML)

ASF-defined tables are implemented as logical records. Application programs using DML statements can access tables through the **Logical Record Facility (LRF)**. The LRF DML commands OBTAIN, STORE, MODIFY, and ERASE are used to manipulate data in tables defined through ASF.

### Build Access Paths

For a program to access a particular table, a predefined access path must exist in the subschema for the DML verb being used. The paths are defined according to the responses contained in the following fields on the Extended Table Definition screen:

- **Display Access**—If YES (default), an OBTAIN path is built, allowing a program to retrieve the table.
- **Load Access**—If YES, a STORE path is built, allowing a program to load data into the table. YES is the default for stored tables. For views, this field is always NO and cannot be changed, though the DBA can define a STORE path for the view by using the subschema compiler.
- **Change Access**—If YES (default), a MODIFY path is built, allowing a program to change data in the table.
- **Erase Access**—If YES, an ERASE path is built, allowing a program to delete data from the table. YES is the default for stored tables. For views, this field is always NO and cannot be changed, though the DBA can define an ERASE path for the view by using the subschema compiler.

### Access Tables from Programs

Data tables can be accessed through LRF DML programs written in COBOL, PL/I, or Assembler.

To access a stored table or view through a DML program, specify the **logical record name**. The logical record name and the **subschema name** can be obtained from the Extended Table Definition screen. The schema name assigned at installation is **IDMSR**.

**Note:** When accessing a data table through a DML program, any selection criteria and/or calculations defined for the table *will* apply.

This product ignores row-level security.

## Application Development System (CA ADS)

### Table Access

ASF-defined tables can be accessed by CA ADS dialogs. Data table access in the CA ADS dialog is accomplished through the Logical Record Facility. The considerations outlined above for using data tables in DML programs also apply when using data tables through CA ADS. Note that only one ASF-defined table can be accessed in each dialog.

### Application Development

Since the application created through ASF is the CA ADS dialog, you can modify the ASF-generated dialog through CA ADS to customize the application. You can also use the **CA ADS application compiler** to combine ASF-created application components into an application system accessed as the CA ADS task.

**Note:** This product ignores row-level security.



# Chapter 13: Basic Screens

---

The Automatic System Facility includes the following basic screens, which are used by all users to define and use data tables:

- Signon screen
- Activity Selection screen
- Table Definition screen
- Column Definition screen
- Data Selection screen
- Data Display screen
- Message screen
- Help screens

This section contains the following topics:

[Signon Screen—SIGN](#) (see page 186)

[Activity Selection Screen—ASEL](#) (see page 188)

[Table Definition Screen—TDEF](#) (see page 191)

[Column Definition Screen—CDEF](#) (see page 196)

[Data Selection Screen—DSEL](#) (see page 200)

[Data Display Screen—DLOD \(or\) DDIS](#) (see page 209)

[Message Screen—MSGS](#) (see page 211)

[Help Screens—HELP](#) (see page 212)

## Signon Screen—SIGN

### Description

The Signon screen enables you to:

- Sign on to the valid catalog for the desired ASF system.
- Sign on as an authorized ASF user.
- Sign on with the appropriate password.
- Sign on to ASF as a different user than the one you are currently signed on as.

### Access

You can access the Signon screen in one of the following ways:

- **Enter the task code ASF from DC/UCF**; the Signon screen is displayed if your DC/UCF signon is not sufficient to access the ASF system.
- **Select the Signon function** from the Activity Selection screen; you can select the function by pressing [PF14].
- **Enter the screen name SIGN** from any other ASF screen.

You can exit the Signon screen in one of the following ways:

- **If signon is successful**, the Activity Selection screen is displayed. You can begin the ASF session as the signed-on user.
- **If signon is unsuccessful**, press [Clear] to leave ASF and return to DC/UCF.
- **If you are already signed on** and make no changes on the Signon screen, press [Clear] to return to the prior screen or enter a screen name in the upper left corner of the screen to move to another screen. You can resume the current ASF session.

**Sample screen**

```

SIGN  CA - Automatic System Facility          CA
DC560002 SELECT NEXT ACTIVITY              *** Signon Screen ***

                                           _ PF1 - Help

                                           ffffffff
                                           ffffffff
                                           fffff      ffff
                                           fffff      ffff
aaaaa      aaaaa      SSSSS      SSSS      fffff      ffff
aaaa      aaaa      SSSS      SSSS      fffff      ffff
aaa      aaaa      SSSS      SSSS      fffff      ffff
                                           ffffffff
aaaaa      aaaaa      SSSSS      SSSS      fffff
aaaaa      aaaaa      SSSSS      SSSS      fffff
aaaa      aaaa      SSSS      SSSS      fffff
aaaaa      aaaaa      SSSS      SSSS      fffff
aaaaa      aaaaa      SSSS      SSSS      fffff
aaaaa      aaaaa      SSSS      SSSS      fffff
aaaaa      aaaaa      SSSS      SSSS      fffff
                                           fffff

User Name: SNELL

Password. :

Catalog..: ASFDICT
    
```

**Fields**

**[PF1] - Help**

Invokes the Help screen.

**User Name**

By default, contains the user ID used to sign on to DC/UCF. You can sign on to ASF as a different user by replacing the default DC/UCF user ID with a different user ID.

**Password**

A dark field used to enter the user's password. The password defaults to the DC/UCF password of the current user. If the DC signon password is not sufficient, or if a different user ID is supplied, a valid password must be supplied.

**Catalog**

Contains the name of the dictionary in which the catalog resides. Optionally, enter the correct catalog name for the ASF system. The catalog can also be set with a DCUF SET DICTNAME command in DC/UCF.

## Activity Selection Screen—ASEL

### Description

The Activity Selection screen presents the activities and tables available to you during an ASF session. The Activity Selection screen enables you to perform the following functions:

- Select an activity
- Name a new table to be defined
- Select an existing table to which you have access
- Browse the table directory

### Access

You can access the Activity Selection screen in one of the following ways:

- **Sign on successfully to ASF.** If your DC/UCF signon is sufficient to access the ASF system, the Activity Selection screen is displayed immediately. Otherwise, you must sign on to ASF using the Signon screen discussed previously. The Activity Selection screen is displayed automatically when your signon is accepted.
- **Press [PA1]** from any other ASF screen.
- **Enter ASEL** in the screen name field of another ASF screen.

You can leave the Activity Selection screen and end the ASF session by pressing [Clear]; this returns you to DC/UCF.

### Sample Screen

```

                                         CA
ASEL  CA - Automatic System Facility      ** Activity Selection **
DC560122 SIGNON COMPLETED, SELECT A TABLE

User Name: SNELL

  _ PF1 - Help                _ PF5 - Select Data          _ PF13 - Query
  _ PF2 - Define Table        _ PF7 - Page Backward       _ PF14 - Signon
  _ PF3 - Load Data          _ PF8 - Page Forward        _ PA1  - Prior Level
  _ PF4 - Display/Change Data _ PF9 - Passkey             _ CLEAR - Leave ASF
                                         _ PF15 - Administrator

Table Name.:
Table Owner: SNELL

                                         Page: 1 of 1

  _ APPLICANT
  _ EMPLOYEES
  _ JOB
  _ JOBS AND APPLICANTS

```

**Fields****User Name**

Displays the name of the user currently signed on.

**[PF1] - Help**

Invokes the Help screen.

**[PF2] - Define Table**

Invokes the Table Definition screen, which enables you to define a new table or to modify or delete an existing table.

**[PF3] - Load Data**

Invokes the Data Display screen for an existing table, which enables you to load data to be stored in the table.

**[PF4] - Display/Change Data**

Invokes the Data Display screen for an existing table, which enables you to display, change, delete, and load data.

**[PF5] - Select Data**

Invokes the Data Selection screen for a generated table, which enables you to select data for display by specifying selection criteria.

**[PF7] - Page Backward**

Causes the previous page of directory listings to be redisplayed. The table directory is in the lower portion of the screen.

**[PF8] - Page Forward**

Causes the next page of directory listings to be displayed. The table directory is in the lower portion of the screen.

**Note:** [PF7] and [PF8] are global PF-keys and can be used on any ASF screen that has paging.

**[PF9] - Passkey**

Invokes the Passkey Function screen, which enables you to select passkey activities.

**[PF13] - Query**

Invokes OnLine Query, which enables you to display a generated table in row and column format. This function applies only if OnLine Query is installed at your site.

**[PF14] - Signon**

Invokes the Signon screen, which enables you to sign on to ASF as a different user or to sign on to a different catalog.

**[PA1] - Prior Level**

Displays the previous level of directory listings.

**[Clear] - Leave ASF**

Returns you to DC/UCF and ends the ASF session. You must return to the Activity Selection screen and press [Clear] to end an ASF session.

**[PF15] - Administrator**

Appears only when the user has administrative authority; invokes the Administrator Control screen.

**Table Name**

Displays the name of the table currently in use in the ASF session. At first signon, this field is blank until a table is accessed. To access a table, enter the name of a new or existing table, as described below.

**Table Owner**

Displays the user ID of the table owner. By default, this is the user ID of the signed-on user. You can replace the contents of this field with another user ID, as described below, to list or use tables owned by others.

**Page: \_\_\_ of \_\_\_**

Displays the current page number and the total number of pages in the table directory. You can move to a particular page by specifying its page number, page forward by using [PF8], and page backward by using [PF7].

**Table Directory**

The Table directory lists the tables available for your use. At first signon, the tables listed are those owned by the user currently signed on to ASF. If a different table owner is specified in the Table Owner field and you have a BROWSE passkey for the owner name, the tables owned by that user are listed. To access a table, type any character in the response field next to the table name.

**Note:** In CA-ICMS installations, the table directory can also include **folders**, which are indicated in the table directory by three trailing periods (...). To list the contents of a folder, select the folder from the table directory and press [Enter]. The folder name displays in the Table Name field and the tables contained in the folder are listed in the table directory.

### Special Considerations

Use the table directory, the Table Name field, and the Table Owner field to list and select tables, as follows:

- **To develop a new table** in this ASF session, enter the name of the new table in the Table Name field. Table names can contain 1 to 32 characters. Any character is valid, except a period (.), underscore (\_), accent grave (&grave.), and backslash (\). A cent sign (&cent;) is allowed, but not recommended. Table names can include embedded spaces, but cannot begin with a space. A table name cannot be the same as the owner name.
- **To process an existing table that you own**, enter the name of the table in the Table Name field or select the table from the table directory.
- **To list the directory of another user**, enter the owner's name in the Table Owner field. The Table Name field must be blank. Note that you must have a BROWSE passkey for the owner name in order to list the directory.
- **To process an existing table owned by another user**, select the table from the user's directory or enter the table name in the Table Name field and the owner's name in the Table Owner field. Note that you must have the appropriate passkeys in order to access the table.
- **To list the contents of a folder**, select the folder from the directory or enter the folder name in the Table Name field. Folders are indicated in the table directory by three trailing periods (...). (Do not enter the periods after the folder name.) Press [Enter] to list the tables contained in the folder. (This applies only when CA-ICMS is installed.)
- **To redisplay the prior level of folders**, press [PA1].
- **To redisplay the directory** after a table has been selected, press [PA1]. This cancels the table selection. If no table has been selected, but you are listing the directory of another user, [PA1] enables you to redisplay your own directory.

## Table Definition Screen—TDEF

### Description

The Table Definition screen allows you to:

- Define a new table
- Review or update the definition of an existing table
- Delete a table

To define a new table, you specify the table type, either **view** or **stored**. If the table is a view, you can specify table derivation information. You can identify one or two source tables used to derive the view and specify the conditions used to join two source tables.

**Note:** If the view is derived from more than two source tables or from one or more records, specify the derivation information on the Extended Derivation screen, discussed in the chapter "Extended Screens".

**Access**

You can access the Table Definition screen in the following ways:

- **Select the Define Table function** on the Activity Selection screen; you can select the function by pressing [PF2].
- **Enter TDEF in the screen name field** of any screen.

**Sample Screen**

```
TDEF  CA - Automatic System Facility          CA
DC560005 MODIFY AND/OR SELECT NEXT ACTIVITY  ** Table Definition **

      _ PF1 - Help                          _ PF4 - Extended Table Definition
      _ PF2 - Define Columns                 _ PF5 - Delete Table Definition
      _ PF3 - Generate                       _ PF6 - Message Screen

Table Name.: APPLICANT
Table Owner: SNELL                          Defn Number: 416
View/Stored: STORED                         Status.....: REDEFINITION
Comments...:

                                Table Derivation

Source Table #1
Table Name.:
Table Owner:

Source Table #2
Table Name.:
Table Owner:

Column #1:
Column #2                                Where Column #1 EQ Column #2
```

**Fields**

**Table Name**

Displays the name of the table specified on the Activity Selection screen.

**Table Owner**

Displays the name of the table owner specified on the Activity Selection screen.



**View/Stored**

Enter the table type VIEW or STORED. The default value for this field is STORED. To define a view, specify VIEW. Enter a minimum of one character (V or S) to indicate the table type. The table type can only be changed when the table's status is NEW (see the Status field below).

**Comments**

Enter an optional short (up to 40 characters) description of the table. If DBCS is installed, this field allows both EBCDIC and DBCS characters.

**Defn Number**

Displays the table definition number generated by the system when the table is defined.

This field can be set by a DBA (or someone with administrative authority), before ASF assigns a number. If ASF has already set a number, it cannot be changed. If the definition number is set to a number other than the default number that ASF would use, ASF will use the skipped numbers for subsequent tables **in reverse order**. ASF will start with the number preceding the one specified in this field, and work backwards until the skipped numbers are used up. For instance, if the last table's definition number was 120, and you changed the definition of this table to 140, the next table generated will have a table definition number of 139.

**Status**

Shows the current status of the table, as follows:

- **NEW** indicates that the table has been named, but not defined. If defining a view, you must change the View/Stored field to VIEW while the table has a status of NEW. Pressing [Enter] or any valid PF-key causes the table to be added to the catalog, changing the status from NEW to UNGENERATED.
- **UNGENERATED** indicates that the table has been defined, but is not yet available for use in an application.
- **GEN-PHASE-1/GEN-PHASE-2/GEN-PHASE-3** indicates that the generation of the table was interrupted and the table should be generated again. The table can only be generated or deleted. You cannot modify the table definition if a GEN-PHASE status applies, with one exception: if the generation failed due to improper space allocations, you can modify the Primary Number Of Rows, Secondary Number Of Rows, and Maximum Number Of Rows fields on the Extended Table Definition screen.
- **GENERATED** indicates that the table can be used.

- **REDEFINITION** indicates that the definition of a generated table has been modified, but the table must be regenerated for the modifications to take effect. You can use a table with this status, but you cannot perform run-time selection (DSEL) or access the table through CA OLQ until you have generated the table again.
- **MUST REGENERATE** indicates that the table must be regenerated before it can be used in an application. This status usually applies to views when changes have been made to the derivation source tables (the tables that make up the view).

**[PF1] - Help**

Invokes the Help screen.

**[PF2] - Define Columns**

Invokes the Column Definition screen, which allows you to define, modify, or delete columns for the table.

**[PF3] - Generate**

Creates the application components that allow you to use the table.

**[PF4] - Extended Table Definition**

Invokes the Extended Table Definition screen, which allows you to specify additional characteristics for the table and displays more information about the table.

**[PF5] - Delete Table Definition**

Deletes the table definition and its application components. For a stored table, this action erases from the database all data stored in the table. When this function is selected, you are prompted to press [Enter] to delete the table.

**[PF6] - Message Screen**

Indicates that additional error messages can be displayed on a separate screen. The message display line will instruct you to press [PF6] when there are error messages to be displayed.

**Table Derivation**

Enter the source table names and criteria to create a new table from one or two existing tables. Note that you must hold a COPY passkey for the source tables to perform table derivation. Through table derivation, you can perform the relational operations **project**, **select**, and **join**. You can specify a single source table and derive a new table by choosing columns (projecting) or by choosing specific rows in the table (selecting). Alternatively, you can specify two source tables and join them on a common column value. From the joined tables you can then choose columns and select rows.

Define a table through a simple table derivation by completing the response fields under the Table Derivation heading, as follows:

- **Source Table #1** first (or only) source table, as follows:
  - **Table Name** of the table to be used as the first source table.
  - **Table Owner**—Enter the name of the table's owner. If you own the table, you need not specify the table owner.

If you are deriving a table from only one source table, you do not have to enter any other information for the simple table derivation.

- **Source Table #2**—Enter information about the second source table (if any) in the simple table derivation, as follows:
  - **Table Name**—Enter the table name (or access path) of the table to be used as the second source table.
  - **Table Owner**—Enter the name of the table's owner. If you own the table, you need not specify the table owner.

Complete the following fields to specify the criteria that join the two source tables. Typically, a simple table derivation is based on a column common to both source tables. This column is called the *join column*.

- **Column #1**—Enter the name of the join column contained in derivation table #1. The join column must be related to a column contained in derivation table #2.
- **Column #2**—Enter the name of the join column contained in derivation table #2. This column must be related to Column #1.
- **Where Column #1 EQ Column #2**—Specifies the condition on which the two columns are joined. The two source tables are joined each time the values in the join columns are equal. The default value EQ is the most common condition used to join two tables. Unequal conditions, represented by NE, GT, LT, LE, or GE, can be specified instead of EQ but are not typically used in simple derivations.

**Note:** You can use the Table Definition screen to begin defining a table derivation, then access the Extended Table Derivation screen to identify additional sources and/or join criteria. To access the Extended Table Derivation screen, type **xder** in the screen name field.

## Column Definition Screen—CDEF

### Description

The Column Definition screen allows you to add, modify, and delete columns in data tables. The column definitions are used to store data in the table and to format a screen for displaying the table online.

**Note:** While there is virtually no limit to the number of columns you can define for a table, the screen format generated to display the table is limited by the size of the terminal screen.

Use the Column Definition screen to define columns, as follows:

- **If you are creating a new stored table**, supply the column names, length, type, and other characteristics to be included in the definition.
- **If you are deriving the table from existing tables**, modify the definitions of the source table columns on the Column Definition screen. You can also delete columns you do not need and add new columns if you wish. When you delete columns, they are deleted from the table definition and cannot be added back. When you add columns, they are work columns and are not used to store data.
- **If you are modifying the definition of an existing table**, you can change column names and/or characteristics and delete entire column definitions. The modifications you make will automatically be reflected in the storage and display formats for the columns when the table is regenerated.

**Note:** If you delete a column from a table, be sure to remove any references to that column in calculations or selection criteria defined for the table.

### Access

You can access the Column Definition screen in the following ways:

- **Select the Define Columns function** on the Table Definition screen; you can select the function by pressing [PF2].
- **Enter CDEF in the screen name field** of any screen.

**Sample Screen**

Column Name	Width	Type	Unique Key	Display Seq	Ext Screen
NAME	30	TEXT	X	1	-
STREET	30	TEXT	-	2	-
CITY	25	TEXT	-	3	-
STATE	2	TEXT	-	4	-
ZIP CODE	5	TEXT	-	5	-
PHONE NUMBER	13	TEXT	X	6	-
POSITION APPLIED FOR	30	TEXT	X	7	-
SALARY REQUIREMENT	7	CURRENCY	-	8	-
EVALUATION: STRONG,WEAK,MEDIUM	6	TEXT	-	9	-

**Fields****Table Name**

Displays the name of the table, as specified on the Activity Selection screen.

**[PF1] - Help**

Invokes the Help screen.

**[PF2] - Extended Column Definition**

Invokes the Extended Column Definition screen, which allows you to specify additional characteristics for the column and displays more information about the column.

**[PF3] - Extended Selection Definition**

Invokes the Extended Selection Definition screen, which allows you to select, at definition time, the data to be included in the table.

**[PF4] - Extended Key Definition**

Invokes the Extended Key Definition screen, which allows you to establish indexes for a table.

**Page: \_\_\_ of \_\_\_**

Displays the current screen page number and the total number of pages in the page set. You can move to a particular page by specifying its page number, page forward by using [PF8], and page backward by using [PF7].

### Column Name

Enter, modify, or delete the column name, as follows:

- **To create names for new columns**, enter a 1- to 32-character alphanumeric value. Special characters, including embedded blanks and hyphens, are allowed. This field also allows DBCS characters, with the restriction that the stored column name, generated by ASF from the displayed column name, must be unique within the table definition.
- **To modify existing column names**, replace existing column names with new column names.
- **To delete existing columns**, erase the column names by spacing over the existing names or by using the ERASE EOF key. Erasing the column name causes the column definition to be deleted. Note that if you delete a column in a view, you *cannot* restore the column by reentering the column name. Once deleted, the column is no longer part of the view definition.

**Note:** You cannot delete a column from a view if the column is defined as an index key in the source table.

### Width

Enter the width of each column. The width is the number of characters or digits the column is to contain. If the column type is CURRENCY (see the Type field below), the minimum column width specified must be 2, for the two decimal places. You cannot change the width of derived columns in views.

**Note:** When defining the width of numeric columns, you are specifying the number of digits the column can contain, not the length of the response field needed to display the column on the formatted screen. The external picture of the column, which is displayed in the Display Format field of the Extended Column Definition screen, determines how long the response field for the column will be.

When considering the format of the screen that the system will generate for the table, keep in mind that the width determines how many lines on the screen a column requires to display data. Assuming an 80-character screen, the first 46 characters of data can be displayed on the same line as the column name. Columns wider than 46 characters will be displayed on more than one line on the formatted Data Display screen.

### Type

Enter the column type as TEXT, NUMERIC, or CURRENCY. You can enter a 1-character response (T, N, or C) for each column, as follows:

- **TEXT** alphanumeric, or graphic (DBCS) values. TEXT is the default column type; if you do not specify anything in the Type column, the system will supply TEXT.
- **NUMERIC** numeric values. To allow for decimal point entry in a numeric value (other than currency), use the Extended Column Definition screen.
- **CURRENCY** monetary values. When you specify CURRENCY, the system automatically inserts a decimal point followed by two decimal places.

You cannot change the column type once the column has been added.

### Unique Key

Designates one or more columns as an index key for the table. The system creates an index using the unique key to store the rows of the table. By default, the unique key is used to display the rows of a table in ascending order. The display sequence can be changed by using the Extended Key Definition screen. To specify a column or columns as a unique key, enter any character in the Unique Key field for the chosen column(s). You can define a unique key for stored tables only.

**Note:** The data contained in the key column(s) must be unique to the table; duplicate data is not allowed. The key column(s) is used to sort the rows, in ascending order, for storage and display.

### Display Seq

Specifies the order in which columns are to appear when the table is displayed online through the generated application. The following considerations apply to the use of this field:

- Number the columns in the order you want them to appear on the screen. Each sequence number must be unique; duplicate numbers are ignored.
- Erase the display sequence number for any column you do not want displayed (or specify a display sequence of 0 (zero)).
- To leave a blank line between columns when they are displayed, number the display sequence by increments of two (for example, 1, 3, 5, and so on).
- You can enter the display sequence numbers for all columns when you add the column definitions, or you can make changes to the display sequence of individual columns during a subsequent modification.

The default display sequence is the order in which the columns are defined.

**Note:** Although all columns are given a display sequence number, whether specified by you or automatically assigned by the system, the number of columns actually displayed on the formatted screen depends on the number of available lines on the terminal screen. Columns that do not appear on the formatted screen do exist in the table, but they cannot be displayed.

Derived columns that result from a second-level (or greater) OCCURS clause cannot be displayed through ASF. They are automatically assigned a display sequence of 0 (zero), which cannot be changed. For example, the following column definitions were derived from a table:

```
REGION.1  
REGIONAL-MANAGER.1  
SALES-MANAGER.1.1  
SALES-MANAGER.1.2  
REGION.2  
REGIONAL-MANAGER.2  
SALES-MANAGER.2.1  
SALES-MANAGER.2.2  
REGION.3  
REGIONAL-MANAGER.3  
SALES-MANAGER.3.1  
SALES-MANAGER.3.2
```

The SALES-MANAGER columns cannot be displayed because they are derived from a field that has a second-level OCCURS clause. Online mapping, which creates the screen format for the ASF-generated application, supports first-level OCCURS clauses only. The SALES-MANAGER columns are part of the table definition, but cannot be displayed online through the ASF-generated application.

#### **Ext Screen**

Selects a column for display through the Extended Column Definition screen. Enter any character in the Ext Screen response field for the columns you want to display and press [PF2]. The Extended Column Definition screen is displayed for the first column selected. Continue to press [PF2] to display the extended definitions for the remaining selected columns.

To display the extended definitions for all columns, leave the Ext Screen field blank for all columns and press [PF2]. The Extended Column Definition screen is displayed for the first column defined on the Column Definition screen. Continue to press [PF2] to display the extended definition for each remaining column on the page. To display extended definitions for columns on a subsequent page, you must go to that page of the Column Definition screen and press [PF2].

## **Data Selection Screen—DSEL**

#### **Description**

The Data Selection screen is used to select rows from a table for display through the ASF-generated application.



### Access

The Data Selection screen can be accessed for any table with a status of GENERATED. You can access the Data Selection screen in the following ways:

- **Select the Select Data function** from the Activity Selection screen; you can select the function by pressing [PF5].
- **Enter DSEL in the screen name field** of any screen.

To display selected rows, enter selection criteria and press [Enter]. When ASF saves the criteria you have specified, press [Enter] again to access the Data Display screen, which displays the selected rows.

### Sample Screen

```

DSEL  CA - Automatic System Facility          CA
DC560603 ENTER SELECTION CRITERIA          ** Data Selection **
Table Name: APPLICANT
      _ PF1 - Help      _ PF2 - Message Screen      Page: 1 of 1

  ID      Column Name      Selection Criteria
0001 NAME
0002 STREET
0003 CITY
0004 STATE
0005 ZIP CODE
0006 PHONE NUMBER
0007 POSITION APPLIED FOR
0008 SALARY REQUIREMENT
0009 EVALUATION: STRONG,WEAK,MEDIUM

Additional Selection Criteria
$5 OR $6 OR #6 MATCHES '617'
```

### Fields

#### Table Name

Displays the name of the table as specified on the Activity Selection screen.

#### [PF1] - Help

Invokes the Help screen.

#### [PF2] - Message Screen

Invokes the display of additional error messages on a separate screen. The message display line will tell you when there are error messages to be displayed.

**Page: \_\_\_ of \_\_\_**

Displays the current screen page number and the total number of pages in the page set. You can move to a particular page by specifying its page number, page forward using [PF8], or page backward using [PF7].

**ID**

Displays the column ID number that can be used to reference the column in selection criteria or in calculations defined on the Extended Column Definition screen.

**Column Name**

Displays the name of each column defined for the table.

**Selection Criteria**

Enter fixed-format expressions to select rows based on column values. Fixed-format expressions are explained below. Enter the expression next to the name of the appropriate column(s). If you enter selection criteria for more than one column, you are indicating that only those rows that meet all criteria for all columns are to be displayed.

**Additional Selection Criteria**

Enter freeform expressions to select rows based on compound conditions. Freeform expressions can include compound conditions using the logical operators AND, OR, NOT, &, |, and ~. You can use parentheses to affect the order in which logical operators are evaluated. Freeform expressions can be used in addition to, or instead of, fixed-format expressions to specify selection criteria. When used in conjunction with fixed-format expressions, freeform expressions can be used to override the implied AND connecting fixed-format expressions. Freeform expressions are discussed below.

**Special Considerations**

To select rows for display, specify selection conditions in the Selection Criteria and/or the Additional Selection Criteria fields. The Selection Criteria field accepts fixed-format expressions; the Additional Selection Criteria field accepts freeform expressions:

- **Fixed-format expressions** allow you to define selection criteria on a column-by-column basis, using simple logical expressions (including arithmetic expressions) that contain a single operator (such as EQ, GT, or MATCHES). With fixed-format expressions, you can define criteria for more than one column. The system then selects rows where all expressions are true (that is, using an implied AND as a conjunction between each fixed-format expression). `:tref refid=ffsel.` illustrates the format of fixed-format expressions used to specify selection criteria.
- **Freeform expressions** allow you to define selection criteria using compound logical expressions (including arithmetic expressions) that contain several operators connected with AND and OR. Freeform expressions also enable you to specify criteria using NOT. Freeform expressions can be used in addition to, or instead of, fixed-format expressions.

The following rules govern all selection criteria, both fixed-format and freeform:

- **To specify an alphanumeric string**, enclose the string in quotation marks. If the string contains a quotation mark, double the embedded quotation mark; for example, 'O' 'BRIEN'.
- **To specify a column name containing special characters** (including hyphens, spaces, and periods), enclose the column name in parentheses following the literal COLUMN, which can be abbreviated to COL; for example, COL(LATE PAYMENT COUNT).
- **To evaluate a column entry character by character**, use a mask to specify the exact or generic character to appear in each position. Use the following mask characters to specify generic characters:
  - \* to represent any character
  - @ to represent any alphabetic character
  - # to represent any numeric character

A string containing mask characters can be used with the operators CONTAINS and MATCHES and must be enclosed in quotation marks. Column positions not included in the mask are assumed to contain any character.

- **To refer to a column by column ID number**, precede the column ID number by # (pound sign); for example, #4.
- **To select column entries based on a computed value**, enter an arithmetic expression after the conditional operator. Arithmetic expressions can include:
  - Numeric values
  - References to one or more columns in the table, including the column for which the calculation is being defined. References can be designated as follows:
    - By entering the column name
    - If the column name includes special characters, by entering COL(*column-name*)
    - By entering the ID assigned to the column on the Extended Selection Definition screen, preceded by a pound sign: #*column-id*
  - Arithmetic operators:

Operator	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division

- Parentheses to separate operations or alter the order of evaluation

ASF follows the standard order of precedence when evaluating arithmetic expressions. The order is multiplication and division from left to right, then addition and subtraction from left to right, with operations in parentheses resolved first. If more than one arithmetic expression has been defined in the selection criteria, they are evaluated in the order in which the columns were defined.

- **To specify a range of values** in a fixed-format expression, use the keyword THRU to separate the first and last values in the range.
- **To specify a list of values** in a fixed-format expression, separate each value in the list by a comma, which acts as an implied OR.
- **To refer to a fixed-format expression in a freeform expression**, precede the column ID number by \$ (dollar sign); for example, \$4.

**Fixed format selection criteria:**

First Operand	Conditional Operator	Second Operand
Column name (Provided by the system)	EQ (assumed) = NE GT > LT < GE LE CONTAINS MATCHES	Expression*
	EQ = NE	Expression THRU expression Expression, expression...

\*Expression can be any of the following:

- Quoted string
- Quoted string including mask characters (valid for CONTAINS and MATCHES only)
- Decimal number with optional sign

- Name of another column in the table
- Column ID number in the form *#column-id*
- Arithmetic expression

**Note:** Only the EQ(=) and NE operators can be used with a range (represented by THRU) or with a list of values joined by implied ORs (represented by commas).

**Freeform selection criteria:**

First Operand	Logical Operator	Second Operand
	AND (&) OR ( )	
* NOT (-)		* NOT (-)
<i>\$column-id</i>		<i>\$column-id</i>
comparison		comparison

\*NOT is optional

Using *\$column-id* as the comparison allows you to override the implied AND invoked when multiple fixed-format expressions are specified. For example, if you have specified fixed-format criteria for columns #1, #3, and #6, using fixed-format criteria alone implies that all three conditions must be true for a row to be selected. By using freeform selection criteria, you can specify \$1 OR \$3 as additional criteria. Then, the condition for column #1 or column #3 and the condition for column #6 must be true for the row to be selected.

**Valid comparisons for freeform selection criteria:**

First Operand	Conditional Operator	Second Operand
#Column id		Expression*
Column name	EQ	
	=	
	NE	
	GT	
	>	
	LT	
	<	
	GE	
	LE	
	CONTAINS	
	MATCHES	

\*Expression can be any of the following:

- Quoted string
- Quoted string including mask characters (valid for CONTAINS and MATCHES only)
- Decimal number with optional sign
- Name of another column in the table
- Column ID number in the form #*column-id*
- Arithmetic expression

The following examples illustrate the use of fixed-format and freeform expressions to specify selection criteria.

## Examples

### Selecting from a Range of Values

To select customers with sales in the \$25,000.00 to \$30,000.00 range, specify selection criteria as follows:

ID	Column Name	Selection Criteria
0001	CUSTOMER NAME	
0002	STATE	
0003	CUSTOMER CLASS	
0004	AMOUNT OF SALES	25000 THRU 30000
0005	EARLY PAYMENT COUNT	
0006	ONTIME PAYMENT COUNT	
0007	LATE PAYMENT COUNT	

This expression selects all rows with a sales amount between \$25,000.00 and \$30,000.00

### Selecting from a List of Values

To select all customers in New England, specify selection criteria as follows:

ID	Column Name	Selection Criteria
0001	CUSTOMER NAME	
0002	STATE	'MA', 'ME', 'VT', 'CT', 'RI', 'NH'
0003	CUSTOMER CLASS	
0004	AMOUNT OF SALES	
0005	EARLY PAYMENT COUNT	
0006	ONTIME PAYMENT COUNT	
0007	LATE PAYMENT COUNT	

This expression selects all rows with a state column containing the abbreviation of a New England state.

### Selecting Based on Equal Values

To select customers named O'Brien that have an AAA class rating, specify selection criteria as follows:

ID	Column Name	Selection Criteria
0001	CUSTOMER NAME	'O''BRIEN'
0002	STATE	
0003	CUSTOMER CLASS	'AAA'
0004	AMOUNT OF SALES	
0005	EARLY PAYMENT COUNT	
0006	ONTIME PAYMENT COUNT	
0007	LATE PAYMENT COUNT	

These expressions select all rows with the name O'Brien and a customer class of AAA.

### Selecting Based on Relative Values

To select customers who consistently make late payments, specify selection criteria as follows:

ID	Column Name	Selection Criteria
0001	CUSTOMER NAME	
0002	STATE	
0003	CUSTOMER CLASS	
0004	AMOUNT OF SALES	
0005	EARLY PAYMENT COUNT	
0006	ONTIME PAYMENT COUNT	
0007	LATE PAYMENT COUNT	GE #5 + #6

This expression selects all rows in which the LATE PAYMENT COUNT is greater than or equal to the EARLY PAYMENT COUNT (referenced by its column ID number 5) plus the ONTIME PAYMENT COUNT (referenced by its column ID number 6).

### Selecting Based on a Contained Value

To select all customers who have one of the class A ratings, specify selection criteria as follows:

ID	Column Name	Selection Criteria
0001	CUSTOMER NAME	
0002	STATE	
0003	CUSTOMER CLASS	CONTAINS 'A'
0004	AMOUNT OF SALES	
0005	EARLY PAYMENT COUNT	
0006	ONTIME PAYMENT COUNT	
0007	LATE PAYMENT COUNT	

This expression selects the following customer classes: A, AA, AB, AAA, and BAA.

### Selecting Based on Matching Values

To select all customers whose names begin with O' (such as O'Brien), specify selection criteria as follows:

ID	Column Name	Selection Criteria
0001	CUSTOMER NAME	MATCHES 'O' '@'
0002	STATE	
0003	CUSTOMER CLASS	
0004	AMOUNT OF SALES	
0005	EARLY PAYMENT COUNT	
0006	ONTIME PAYMENT COUNT	
0007	LATE PAYMENT COUNT	

This expression selects only rows that contain a customer name beginning with O'. The mask character @ (representing any alphabetic character) is used to delimit the doubled quotation marks.

### Using Freeform Selection Criteria Alone

To select rows based on an OR condition, specify additional selection criteria as follows:

ID	Column Name	Selection Criteria
0001	CUSTOMER NUMBER	
0002	NUMBER OF SALES	
0003	AMOUNT OF SALES	

Additional Selection Criteria

COL(NUMBER OF SALES) GT 500 OR COL(AMOUNT OF SALES) GT 5000

This expression selects all rows with either more than 500 sales or more than \$5,000 in sales. The criteria can also be specified using the column ID numbers rather than the column names, as follows:

Additional Selection Criteria

#2 GT 500 OR #3 GT 5000



**Using Freeform Selection Criteria with Fixed-Format Expressions**

The selection condition in the above example can also be specified as follows:

ID	Column Name	Selection Criteria
0001	CUSTOMER NUMBER	
0002	NUMBER OF SALES	GT 500
0003	AMOUNT OF SALES	GT 5000

Additional Selection Criteria

\$2 OR \$3

Note that the OR in the freeform expression overrides the implied AND in the fixed-format expressions.

## Data Display Screen—DLOD (or) DDIS

**Description**

A Data Display screen is formatted by ASF for each data table for which an application is generated. The Data Display screen lists the columns in the table that were included in the screen display, along with response fields that allow you to display, load, change, and remove data.

ASF formats the Data Display screen for a table by using a model map. Though the model used at your site may have variations, the model generally used allows 32 characters for each column name and begins each column name on a new line; response fields for the columns are allocated based on the width (or external picture) of the column. When formatted, the Data Display screen lists as many columns and their appropriate response fields as can be displayed on the terminal screen. If the Data Display screen formatted by ASF is inadequate for displaying a particular table, you can modify the Data Display screen manually by using the online mapping facility.

**Access**

You can access the Data Display screen for a particular table in the following ways:

- **Select the Load Data function** from the Activity Selection screen; you can select the function by pressing [PF3]. When accessed through the Load Data function, the Data Display screen displays the column names so that you can enter new data online, one row at a time.
- **Select the Display/Change Data function** from the Activity Selection screen; you can select the function by pressing [PF4]. When accessed through the Display/Change Data function, the Data Display screen displays the column names and one row of data at a time. You can display, add, change, and remove data from the table.
- **Specify selection criteria on the Data Selection screen** and press [Enter]. The Data Display screen is displayed automatically. When accessed through the Data Selection screen, the Data Display screen has the Display/Change Data function, but displays only those rows that meet the selection criteria.
- **Enter DLOD or DDIS in the screen name field** of any screen. When you enter DLOD, the Data Display screen has the Load Data function; when you enter DDIS, the Data Display screen has the Display/Change Data function.

To leave the Data Display screen, press [Clear]. This returns you to the Data Selection screen if the Data Display screen was accessed from the Data Selection screen, or to the Activity Selection screen if the Data Display screen was accessed from any other ASF screen. Note that the Data Display screen is part of the application created for the table and is not actually an ASF screen. Therefore, the Data Display screen does not have a screen name field that can be used to move to any ASF screen. When you leave the Data Display screen, you must return to the prior ASF screen.

**Sample screen**

PF1 = ADD; PF2 = CHANGE; PF3 = DELETE; ENTER = NEXT; CLEAR = EXIT	
JOB	
JOB NUMBER	B359
DEPARTMENT	BILLING
TITLE	BILLING CLERK
SALARY	10000.00
CURRENT STATUS (FILLED OR OPEN)	OPEN

**Fields**

The fields on the Data Display screen vary according to the model used to format the screen and the columns included in the individual table. Generally, the Data Display screen contains the following information:

- **Heading information** is displayed according to the screen model in use at your site.
- **Message text** is displayed, providing instructions and information on the use of the Data Display screen.
- The **table name** of the specified table is displayed.

The remainder of the screen contains a sequential list of columns in the table. If the Display/Change function applies, the screen will also display one row of data. No data is displayed for the Load Data function.

## Message Screen—MSGS

**Description**

The Message screen lists informative messages about ASF processing. At times, you may be instructed to access the Message screen to review error conditions.

**Access**

You can access the Message screen in one of the following ways:

- **Select the Message Screen function** from the Table Definition screen ([PF6]), Data Selection screen ([PF2]), or Extended Selection Definition screen ([PF2]).
- **Enter MSGS in the screen name field** of any screen.

To leave the Message screen and return to the prior screen, press [Clear].

### Sample Screen

```
MSGSG  CA - Automatic System Facility          CA
                                         ** Message Screen **
                                         _ PF2 - Print Messages Page: 1 of 1
DC562100 SYNTAX ERROR IN $00002
DC562009 COLUMN NAME NOT FOUND
DC562101 (E) - SYNTAX ERRORS WERE DISCOVERED BY IDMSREXP
```

### Fields

#### [PF2] - Print Messages

Initiates printing of the displayed messages if a printer is associated with the CA IDMS/DC or CA IDMS UCF system. Press [PF2] to print the page of the screen currently displayed; you must press [PF2] for each page.

Page: \_\_ of \_\_

Displays the current screen page number and the total number of pages in the page set. You can move to a particular page by specifying its page number, page forward by pressing [PF8], and page backward by pressing [PF7].

#### The Messages Area

The remainder of the screen lists system messages. Since ASF uses other products and facilities in its processing, the messages listed may be from another product. For example, the Message screen lists messages from the subschema compiler and CA ADS, as well as messages from ASF itself.

## Help Screens—HELP

### Description

The Help screens contain information that explain the structure and use of the Automatic System Facility and its screens. The help text displayed applies to the screen that was current when the Help facility was accessed. Having accessed the Help facility, you can display the help text for any ASF screen by entering the screen name in the HELP FOR field.

### Access

You can access the Help facility in one of the following ways:

- **Select the Help function** from any screen; you can select the function by pressing [PF1].
- **Enter Help in the screen name field** of any screen.

To leave the Help facility, press [Clear] to return to the prior screen.

### Sample Screen

```

HELP  CA - Automatic System Facility          CA
                                           ** Help Screen **
Help For: ASEL                               Page:  1  of 16

-----
ACTIVITY SELECTION HELP

      Overview..... 2
      The activities..... 3
      Selecting and listing tables..... 4
      Screen names and titles..... 6
      ASF overview..... 8

      Return to ASF      Paging      Activity Selection      Refresh
      CLEAR              PF7/8        PA1                    PA2

----- (continued) -----

```

### Fields

#### Help For

Displays the screen name for which help text is currently being displayed. You can change the screen name to display the help text for another ASF screen.

#### Page: \_\_ of \_\_

indicates the number of pages in the help text. You can page through Help by pressing [PF7] and [PF8] or by changing the page number.

#### Table of contents

Lists the topics discussed in the help text for the current screen. Page numbers are provided so you can go directly to a particular topic.

#### A brief explanation on how to use Help

Provided near the bottom of the first page.

#### The global PF-keys for ASF

Listed at the bottom of the first page.

#### Screen-specific help text

Provided on pageable screens.

**Screen names and titles**

Listed at the end of the screen-specific help text. The titles of all ASF screens and their 4-character screen names are listed. You can go directly to any ASF screen by entering the screen name in the upper left corner of any screen.

**An overview of ASF**

Included at the end of each help text.

# Chapter 14: Passkey Screens

---

The Automatic System Facility includes several passkey screens, which are used by all users to grant access to data tables. This section includes a discussion of passkey security in ASF and describes the passkey screens used to assign, delete, and review passkeys. This section covers the following topics:

- Passkey security
- Passkey Function screen
- Give Passkeys screen
- Delete Passkeys screen
- Show Passkeys Given screen
- Show Passkeys Held screen

This section contains the following topics:

[Passkey Security](#) (see page 216)

[Passkey](#) (see page 220)

[Give Passkeys Screen—GPAS](#) (see page 222)

[Delete Passkeys Screen—DPAS](#) (see page 223)

[Show Passkeys Given Screen—PASG](#) (see page 225)

[Show Passkeys Held Screen—PASH](#) (see page 227)

## Passkey Security

**Passkeys** determine access to data tables. Passkey security is implemented at two levels, as follows:

- **Passkey authorization** determines what you can do with data tables through ASF. By default, ASF users have full passkey authorization, enabling them to create data tables for themselves and be given access to tables owned by others. The ASF administrator can limit the passkey authorization of an individual user through the User Limits and Defaults screen (see the chapter "Administrative Functions of ASF" and the chapter "Administrative Screens"). For example, if the ASF administrator revokes your CREATE passkey authorization, you are not able to define and generate tables through ASF, either for yourself or for other users.
- **Passkey assignments** determine what other users can do with your tables and what you can do with tables owned by others. Table ownership gives you the right to grant or deny another user access to your table. The only way another user, even an ASF administrator, can access a table you own is if you explicitly assign the user a passkey to the table.

Three types of passkeys can be assigned, as follows:

- **Data access passkeys** allow access to data tables and table definitions.
- **Catalog access passkeys** allow access to the catalog, which is the directory of users and tables.
- **Predefined passkey combinations** combine data and catalog access passkeys, enabling you to specify useful combinations of data and catalog access with a single passkey.

Passkey authorization applies only to data access passkeys. When authorization for a passkey is revoked, assignments of that passkey are not honored. For example, if you are not authorized for the ADD passkey, you cannot add data to tables. If another user assigns you an ADD passkey for one of his tables, the passkey is not honored; you cannot add data to your own tables or to another user's tables.

Each of the three types of passkeys that can be assigned through ASF is discussed separately below.



## Data Access Passkeys

**Data access passkeys** govern access to tables and table definitions. The following are the seven data access passkeys:

- **Copy** allows you to use a table as a source for derivation.
 

**Note:** You should be cautious when issuing the Copy passkey. Once a user creates a view using your table(s) as a source for derivation, that user can access your table(s) through the view even if the Copy passkey to the table(s) is removed.
- **Create** allows you to define and store tables through ASF.
- **Add** allows you to add new rows of data to a table and modify and delete existing rows.
- **Modify** allows you to modify and delete rows of data in a table.
- **Erase** allows you to delete a table from ASF through the Delete Table Definition function. If the table is stored, the data is erased from the database.
- **Redefine** allows you to change the definition of a table.
- **List** allows you to display a table, both the data and definition.

For practical considerations, some passkeys are assigned implicitly, as follows:

- Holding any data access passkey grants the List passkey. Therefore, you can display the tables associated with the passkey.
- Holding an Add passkey also implies that you hold a Modify passkey for the tables associated with the passkey. If you can add new rows to a table, you can also modify existing rows.
- Holding a Redefine passkey also implies that you hold a Modify passkey for the tables associated with the passkey. If you can change the definition of a table, you can also modify data in the table.
- Holding a Create passkey grants all data access passkeys to the creator of a table for that table, as long as the Create passkey is held.

Data access passkeys can be held for a user, generally allowing access to all of a user's tables, or for an individual table. The following table shows the data access passkeys and how they work when held for users and tables.

### Data Access Passkeys

Passkey	For access to	Function
COPY	User	Derive from any table owned by the user
	Table	Derive from the table specified
CREATE	User	Create a table for the user

Passkey	For access to	Function
ADD	User	Add additional rows of data to any existing table owned by the user
	Table	Add additional rows of data to the table specified
MODIFY	User	Modify rows in any table owned by the user
	Table	Modify rows in the table specified
ERASE	User	Erase any table owned by the user
	Table	Erase the table specified
REDEFINE	User	Redefine any table owned by the user
	Table	Redefine the table specified
LIST	User	List any table owned by the user
	Table	List the table specified

**Note:** Note that List does not imply Copy; therefore, a table with only a List passkey cannot be used for derivation.

## Catalog Access Passkeys

**Catalog access passkeys** govern access to the catalog, which is the directory of users and tables. The following are the two catalog access passkeys:

- **Browse** allows you to access another user's directory or a table in that directory. To access another user's tables you must hold a Browse passkey to the table or the table's owner, as well as data access passkeys to the table or table owner. Holding a Browse passkey for a user allows you to list the user's directory.
- **Management** allows you to assign passkeys for another user or for another user's table. For example, if you hold a Management passkey for user SNELL, you can grant user CROSS access to tables owned by user SNELL. If you hold a Management passkey for one of user SNELL's tables, you can grant user CROSS access to that table. In CA-ICMS installations, the Management passkey allows manipulation of catalog entities and their relationships through CA-ICMS Manager.

## Predefined Passkey Combinations

ASF supplies **predefined passkey combinations** that allow you to specify useful combinations of data and catalog access with a single passkey. The predefined passkey combinations can be given for a user or for a table, as follows:

- **DBA**—Dba passkey When given for a user, DBA allows you to list a user's directory, create tables for the user, and assign passkeys for the user. When given for a table, DBA allows you to assign passkeys for that table and to access the table if any data access passkeys are assigned.
- **Read**—Read passkey When given for a user, Read allows you to list a user's directory and derive views from tables owned by the user. When given for a table, READ allows you to derive a view from the table.
- **Alter**—Alter passkey When given for a user, Alter allows you to add and modify data in tables owned by the user and to modify definitions of tables owned by the user. When given for a table, Alter allows you to add and modify data in the table and modify the table definition. Note that Alter does not include a Browse passkey, which must be assigned separately.
- **Write**—Write passkey When given for a user, Write allows you to list the user's directory and create tables for the user, and gives you complete data access to tables owned by the user. When given for a table, Write gives you complete data access to the table.
- **Update**—Update passkey When given for a user, Update allows you to create tables for the user and gives you complete data access to tables owned by the user. When given for a table, Update gives you complete data access to the table. Note that the Update passkey does not include a Browse passkey, which must be assigned separately.
- **All**—All passkey When given for a user, All allows you to list a user's directory, create tables for the user, and assign passkeys for the user, and gives you complete data access to tables owned by the user. When given for a table, All allows you to assign passkeys for the table and gives you complete data access to the table.

		PASSKEYS								
		COPY	CREATE	ADD	MODIFY	ERASE	REDEFINE	LIST	BROWSE	MANAGEMENT
PREDEFINED PASSKEY COMBINATIONS	DBA		*						*	*
	READ	*							*	
	ALTER			*	*		*			
	WRITE	*	*	*	*	*	*		*	
	UPDATE	*	*	*	*	*	*			
	ALL	*	*	*	*	*	*	*	*	*

The screens used to assign passkeys, delete passkeys, and review passkey assignments are discussed separately below.

## Passkey

Function Screen—PASS

### Description

contains a menu of passkey activities. From the menu, you select whether you want to:

- Give passkeys to a user
- Delete passkeys you have already given to a user
- Display the passkeys you have given to other users
- Display the passkeys other users have given to you

When you make your selection and press [Enter], the appropriate screen is displayed for the activity you have selected.

**Access** You can access the **Passkey Function screen** in one of the following ways:

- **Select the Passkey function** from the Activity Selection screen; you can select the function by pressing [PF9].
- **Enter PASS in the screen name field** of any screen.

You can leave the Passkey Function screen in one of the following ways:

- **Select an activity from the menu and press [Enter]**; the appropriate screen for performing the activity you select will be displayed.
- **Press [Clear]** to return to the prior screen.
- **Enter the screen name of another screen** in the screen name field.

### Sample Screen

```

PASS CA - Automatic System Facility          CA          *** Passkey Function ***
DC422007 SELECT AN ACTION

                                     _ PF1 - Help

Select Action To Be Performed:

    x Give Passkeys
    _ Delete Passkeys
    _ Show Passkeys Given
    _ Show Passkeys Held

```

### Fields

#### [PF1] - Help

Invokes the Help screen.

#### Give Passkeys

Enter any character to give passkeys to a user; ASF displays the Give Passkeys screen, which allows you to assign passkeys.

#### Delete Passkeys

Enter any character to delete passkeys; ASF displays the Delete Passkeys screen.

#### Show Passkeys Given

Enter any character to display all passkeys given for a specified user name or table; ASF displays the Show Passkeys Given screen.

#### Show Passkeys Held

Enter any character to display all passkeys held by a user; ASF displays the Show Passkeys Held screen.

## Give Passkeys Screen—GPAS

### Description

The Give Passkeys screen allows you to give passkeys to a user that allow that user to access one or all of your tables or folders. You can assign passkeys for an individual table or for your user name. You select the type of access you want the user to have.

### Access

You can access the Give Passkeys screen in one of the following ways:

- **Select the Give Passkeys function** from the Passkey Function screen.
- **Enter GPAS in the screen name field** of any other screen.

### Sample Screen

```
GPAS  CA - Automatic System Facility          CA          *** Give Passkeys ***
DC422007 SELECT AN ACTION

                                     _ PF1 - Help

Select Passkey Type(s):

_ Copy          _ Browse          _ DBA
_ Create        _ Management      _ Read
_ Add           _                 _ Alter
_ Modify        _                 x Write
_ Erase         _                 _ Update
_ Redefine     _                 _ All
_ List

Give The Passkey(s) To This User Or Group:

fites

For Access To The Following Entry:

snell

The Owner Of The Entry Is: SNELL
```

**Fields****[PF1] - Help**

Invokes the Help screen.

**Select Passkey Type(s)**

Enter any character next to the passkey type to be given. One or more passkey types can be selected. The passkey types are explained under "Passkey Security" earlier in this section.

**Give The Passkey(s) To This User Or Group**

Enter the name of the user to receive the passkey(s). If CA-ICMS is installed, you can also enter the name of a group or CORP.

**For Access To The Following Entry**

Enter the name of the user or table for which the passkey is given. If CA-ICMS is installed, you can also enter the name or access path of a table, folder, or CORP.

**The Owner Of The Entry Is**

Enter the name of the owner of the entry for which the passkey is given. For example, if you are giving passkeys for access to one of your tables, enter your user name in this field. Since this is the default, you usually do not need to change the value in this field.

## Delete Passkeys Screen—DPAS

**Description**

The Delete Passkeys screen allows you to delete passkeys you have given to another user. You can delete some or all of the passkeys the user holds for your user name or for an individual table.

**Access**

You can select the Delete Passkeys screen in one of the following ways:

- **Select the Delete Passkeys function** from the Passkey Function screen.
- **Enter DPAS in the screen name field** of any other screen.

**Sample Screen**

```
GPAS CA - Automatic System Facility          CA
DC422007 SELECT AN ACTION                    *** Give Passkeys ***

Select Passkey Type(s):                      _ PF1 - Help

_ Copy          _ Browse          _ DBA
_ Create        _ Management      _ Read
_ Add           _                 _ Alter
_ Modify        _                 x Write
_ Erase         _                 _ Update
_ Redefine     _                 _ All
_ List

Give The Passkey(s) To This User Or Group:

fites

For Access To The Following Entry:

snell

The Owner Of The Entry Is: SNELL
```

**Fields**

**[PF1] - Help**

Invokes the Help screen.

**Select Passkey Type(s)**

Enter any character next to the passkey type to be deleted. One or more passkey types can be selected at one time. The passkey types are explained under "Passkey Security" earlier in this section.

**Delete The Passkey(s) From This User Or Group**

Enter the name of the user for which the passkey(s) is deleted. If CA-ICMS is installed, you can also enter the name of a group or CORP.

**For Access To The Following Entry**

Enter the name of the user or table for which the passkey is deleted. If CA-ICMS is installed, you can also enter the name or access path of a table, folder, or CORP.

**The Owner Of The Entry Is**

Enter the name or access path of the owner of the entry for which the passkey is deleted.



## Show Passkeys Given Screen—PASG

### Description

The Show Passkeys Given screen allows you to request a passkey listing for a user or table. On the Show Passkeys Given screen, you enter a tablename or your user name. All users who have access to the specified table or user are displayed on the Passkeys Given screen. You can see who you have given passkeys to and which passkeys were given.

### Access

You can access the Show Passkeys Given screen in one of the following ways:

- **Select the Show Passkeys Given function** from the Passkey Function screen.
- **Enter PASG in the screen name field** of any other screen.

### Sample Screen

```
PASG CA - Automatic System Facility          CA          *** Show Passkeys Given ***
DC422007 SELECT AN ACTION

                                     _ PF1 - Help

Entry Name: applicant
Owner Name: snell

Requested Action:

          Show The Users And Groups Which Have Access To The Entry Which Is
          Specified Above.
```

### Fields

#### [PF1] - Help

Invokes the Help screen.

#### Entry Name

Enter the name of a user or table.

#### Owner Name

Displays your user name. Enter the appropriate owner name if the specified Entry Name is *not* your user name or a table you own.

#### Requested Action

Displays the action you have requested by choosing this screen.

## Passkeys Given Screen—(No screen name)

### Description

The Passkeys Given screen displays the users who have access to the user or table specified on the Show Passkeys Given screen.

### Access

The Passkeys Given screen *can only be accessed from the Show Passkeys Given screen*. To exit from the Passkeys Given screen, press [Clear] to return to the Show Passkeys Given screen.

### Sample Screen

PASG	CA - Automatic System Facility	CA	*** Passkeys Given ***		
List Of Passkeys Given For:					
APPLICANT					
			Passkeys		
			B	L	C
			C	A	M
			R	S	P
			D	O	R
			S	T	Y
			E	D	D
			S	F	N
Entry Name		Entry Type			
WHITE		USER	Y	Y	Y
FORREST		USER	Y		Y

### Fields

#### List Of Passkeys Given For

Specifies the entity named on the Show Passkeys Given screen.

#### Entry Name

Lists the names of users who hold passkeys for the entity named on the Show Passkeys Given screen.

#### Entry Type

Identifies the holder of the passkeys by type. This typically displays USER, but if CA-ICMS is installed, it can also display GROUP or CATALOG.

**Passkeys**

Specifies which passkeys are held for the entity named on the Show Passkeys Given screen. Possession of a passkey is indicated by a Y in the column under the passkey type. Note that the passkey names listed on the screen are read from top to bottom. The abbreviations for the passkey types are listed below:

- BRS (Browse)
- LST (List)
- CPY (Copy)
- CRE (Create)
- ADD
- MOD (Modify)
- ERS (Erase)
- RDF (Redefine)
- MAN (Management)

## Show Passkeys Held Screen—PASH

**Description**

The Show Passkeys Held screen allows you to request a list of the passkeys held by a specified user. You can specify your user name and see all the tables and users to which you have access. If you have a Management passkey for another user, you can specify that user's name and see what passkeys that user holds.

**Access**

You can access the Show Passkeys Held screen in one of the following ways:

- **Select the Show Passkeys Held function** from the Passkeys Function screen.
- **Enter PASH in the screen name field** of any other screen.

### Sample Screen

```
CA
PASH CA - Automatic System Facility *** Show Passkeys Held ***
DC422007 SELECT AN ACTION

_ PF1 - Help

Entry Name: SNELL

Requested Action:

Show The Entries To Which The User Or Group Specified Above Has Access.
```

### Fields

#### [PF1] - Help

Invokes the Help screen.

#### Entry Name

Enter the name of the user whose passkeys are to be displayed. If CA-ICMS is installed, this can also be the name of a group or CORP.

#### Requested Action

Displays the action you have requested by choosing this screen.

## Passkeys Held Screen—(no screen name)

### Description

The Passkeys Held screen displays the passkeys held by a specified user that allow access to a user or a table.

### Access

The Passkeys Held screen *can only be accessed from the Show Passkeys Held screen*. To exit from the Passkeys Held screen, press [Clear] to return to the Show Passkeys Held screen.

**Sample Screen**

PASH CA - Automatic System Facility		CA	*** Passkeys Held ***							
List Of Passkeys Held By:		SNELL								
Entry Name	Entry Type	Passkeys								
		B	L	C	C	A	M	E	R	M
		R	S	P	R	D	O	R	D	A
		S	T	Y	E	D	D	S	F	N
CROSS										
Owner: CORP	USER	Y	Y	Y	Y	Y	Y	Y	Y	Y
FITES										
Owner: CORP	USER	Y								
FORREST										
Owner: CORP	USER	Y	Y	Y	Y	Y	Y	Y	Y	Y
MONTHLY REPORT										
Owner: FITES	DATA TABLE							Y	Y	Y
WHITE										
Owner: CORP	USER	Y	Y	Y	Y	Y	Y	Y	Y	Y

**Fields**

**List Of Passkeys Held By**

Specifies the user named on the Show Passkeys Held screen.

**Entry Name**

Lists the names of the users and tables to which the user named above has access.

**Entry Type**

Lists the type of entry to which access is granted (USER or DATA TABLE). If the entry type is a table, the owner of the table is also listed. If CA-ICMS is installed, the entry type can also be CATALOG, GROUP, or FOLDER.

**Passkeys**

Specifies which passkeys are held for the entry named above. Possession of a passkey is indicated by a Y in the column under the passkey type. Note that the passkey names listed are read from top to bottom. The abbreviations for passkey types are listed below:

- BRS (Browse)
- LST (List)
- CPY (Copy)
- CRE (Create)
- ADD

- MOD (Modify)
- ERS (Erase)
- RDF (Redefine)
- MAN (Management)

# Chapter 15: Row-Level Security Screens

---

The Automatic System Facility includes several row-level security screens, which are used by all users to restrict access to certain rows of their tables. This section includes a discussion of row-level security in ASF and describes the row-level security screens used to assign, delete, and review row-level security. This section covers the following topics:

- Row-level Security
- System Table Maintenance screen
- Row-level Security screen
- Load Row-level Security screen
- Row-level Security Selection screen
- Display/Change Row-level Security screen

This section contains the following topics:

[Row-level Security](#) (see page 231)

[System Table Maintenance Screen—SYST](#) (see page 232)

[Row-level Security Screen—RSEC](#) (see page 233)

[Load Row-level Security Screen—RLOD](#) (see page 235)

[Row-level Security Selection Screen—RSEL](#) (see page 236)

[Display Row-level Security Screen—RDIS](#) (see page 238)

## Row-level Security

Row-level security determines access to rows of data tables. Row-level security is implemented at two levels, as follows:

- **Row-level security authorization** determines what you can do with row-level security through ASF. By default, no users have access to row-level security unless explicitly authorized by a DBA.

The DBA gives users access to row-level security tables which enables the users to assign row-level security to their tables.

- **Row-level security assignment** determines how other users can use your data tables. You can assign security to certain rows of your tables to limit access to those rows. Passkeys are checked first when a user tries to access a table.

## System Table Maintenance Screen—SYST

### Description

The System Table Maintenance screen allows the DBA (or anyone with DBA authority) to access the following tables directly:

- \$UNSTRUCTURED-IDB-OBJECT\$
- \$SECURITY-RUNTIME-TABLE\$
- \$OBJECT-SECURITY-NAME-TABLE\$
- \$SRT-OST-CROSS-REFERENCE\$

### Access

You can access the System Table Maintenance screen in one of the following ways:

- **Select the System Table Maintenance function** from the Administrator Control screen; you can select the function by pressing [PF8].
- **Enter SYST in the screen name field** of any field.

You can leave the System Table Maintenance screen in one of the following ways:

- **Select an activity from the menu and press [Enter]**; the appropriate screen for performing the activity you select will be displayed.
- **Press [Clear]** to return to the prior screen.
- **Enter the screen name of another screen** in the screen name field.

### Sample Screen

```
SYST  CA - Automatic System Facility          CA
DC560727 SELECT A SYSTEM TABLE             *System Table Maintenance*

Table Name:

    _ PF1 - Help                               _ PF4 - Display/Change Data
    _ PF2 - Define Table                       _ PF5 - Select Data
    _ PF3 - Load Data                         _ PF6 - Query

    _ $UNSTRUCTURED-IDB-OBJECT$
    _ $SECURITY-RUNTIME-TABLE$
    _ $OBJECT-SECURITY-NAME-TABLE$
    _ $SRT-OST-CROSS-REFERENCE$
```

### Fields



**Table Name**

Displays the name of the table as specified on the System Table Maintenance screen.

**[PF1] - Help**

Invokes the Help screen.

**[PF2] - Define Table**

Invokes the Table Definition screen, which enables you to define a new table or to modify or delete an existing table.

**[PF3] - Load Data**

Invokes the Data Display screen for an existing table, which enables you to load data to be stored in the table.

**[PF4] - Display/Change Data**

Invokes the Data Display screen for an existing table, which enables you to display, change, delete, and load data.

**[PF5] - Select Data**

Invokes the Data Selection screen for a generated table, which enables you to select data for display by specifying selection criteria.

**[PF6] - Query**

Invokes OnLine Query, which enables you to display a generated table in row and column format. This function applies only if OnLine Query is installed at your site.

## Row-level Security Screen—RSEC

**Description**

The Row-Level Security screen allows you to assign a security name and generic column names to an existing data table.

**Access**

You can access the Row-Level Security screen by typing **RSEC** in the screen name field of any screen.

**Sample Screen**

```
CA
RSEC  CA - Automatic System Facility      ** Row Level Security **
DC560314 ADD/MODIFY SECURITY NAME AND GENERIC COLUMN NAMES
Table Name: APPLICANT
                                           _ PF1 - Help Page: 1 of 1

Security Name: salsec

      Column Name              Generic Column Name
NAME
STREET
CITY
STATE
ZIP CODE
PHONE NUMBER
POSITION APPLIED FOR
SALARY REQUIREMENT
EVALUATION: STRONG, WEAK, MEDIUM
```

**Fields**

**Table Name**

Displays the name of the table specified on the Activity Selection screen.

**[PF1] - Help**

Invokes the Help screen.

**Page \_\_\_ of \_\_\_**

Displays the current screen page number and the total number of pages in the page set. You can move to a particular page by specifying its page number, page forward by using [PF8], and page backward by using [PF7].

**Security Name**

Enter the security name you are assigning to the data table.

**Column Name**

Displays the columns of the table specified in the Table Name field.

**Generic Column Name**

Enter the generic column names you wish to assign to the existing column names.

## Load Row-level Security Screen—RLOD

### Description

Allows you to load rows in the \$SECURITY-RUNTIME-TABLE\$ to specify who has access to the rows of your tables that have row-level security applied to them.

### Access

You can access the Load Row-Level Security screen by typing RLOD in the screen name field of any screen.

### Sample Screen

```

                                     CA
PF1 = ADD;PF2 = CHANGE;PF3 = DELETE;PF4 = VALIDATE;ENTER = NEXT;CLEAR = EXIT
$SECURITY-RUNTIME-TABLE$

GROUP
USER
OWNER
SECURITY-NAME
STATUS
WHERE-CLAUSE

```

### Fields

#### GROUP

Enter the name of the group you want to have access to your rows. You can do this only if CA-ICMS is installed.

#### USER

Enter the name of the user you want to have access to your rows. Keep in mind that the GROUP field and the USER field are mutually exclusive. You cannot make an entry in both fields at once.

#### OWNER

Enter the name of the owner of the table to which you are applying row-level security.

#### SECURITY-NAME

Enter the security name assigned to the table on the Row-Level Security screen.

**STATUS**

Displays the status of the row-level security assignment. This field will be one of the following:

- (I)—Validation Required
- (V)—Validated
- (E)—Error in Validation
- 

**WHERE-CLAUSE**

Enter the selection criteria that defines which rows of the table the user is allowed to access.

## Row-level Security Selection Screen—RSEL

**Description**

The Row-Level Security Selection screen allows you to select certain rows of the \$SECURITY-RUNTIME-TABLE\$ for display.

**Access**

You can access the Row-Level Security Selection screen by typing RSEL in the screen name field of any screen.

**Sample Screen**

```
RSEL CA - Automatic System Facility          CA
DC560603 ENTER SELECTION CRITERIA          *Row Level Security Selection*
Table Name:                                Page: 1 of 1
      _ PF1 - Help      _ PF2 - Message Screen
ID      Column Name      Selection Criteria
0001 GROUP
0002 USER
0003 OWNER
0004 SECURITY-NAME
0005 STATUS              ne 'v'
0006 WHERE-CLAUSE
0007 GROUP-SEQUENCE-NUMBER
0008 TIME-STAMP
0009 DATE-STAMP
0010 WORK-GROUP

Additional Selection Criteria
```

**Fields****Table Name**

Displays the name of the table, \$SECURITY-RUNTIME-TABLE\$.

**[PF1] - Help**

Invokes the Help screen.

**[PF2] - Message Screen**

Invokes the Message screen which displays error messages should you enter something incorrectly.

**Page \_\_\_ of \_\_\_**

Displays the current screen page number and the total number of pages in the page set. You can move to a particular page by specifying its page number, page forward by using [PF8], and page backward by using [PF7].

**ID**

Displays the column ID number that can be used to reference the column in selection criteria.

**Column Name**

Displays the name of each column defined for the table.

**Selection Criteria**

Enter fixed-format expressions to select rows based on column values. Fixed-format expressions are explained in the chapter "Basic Screens". Enter the expression next to the name of the appropriate column(s). If you enter selection criteria for more than one column, you are indicating that only those rows that meet all criteria for all columns are to be displayed.

**Additional Selection Criteria**

Enter freeform expressions to select rows based on compound conditions. Freeform expressions can include compound conditions using the logical operators AND, OR, NOT, &, |, and ~. Freeform expressions can be used in addition to, or instead of, fixed-format expressions to specify selection criteria. When used in conjunction with fixed-format expressions, freeform expressions can be used to override the implied AND connecting fixed-format expressions. Freeform expressions are discussed in the chapter "Basic Screens".

## Display Row-level Security Screen—RDIS

### Description

Allows you to display, load, change, and delete the rows of the \$SECURITY-RUNTIME-TABLE\$ you are allowed to access.

### Access

You can access the Display Row-Level Security screen by typing RDIS in the screen name field of any screen.

### Sample Screen

NO ROW TO DISPLAY	CA
\$SECURITY-RUNTIME-TABLE\$	
GROUP	
USER	
OWNER	
SECURITY-NAME	
STATUS	
WHERE-CLAUSE	

### Fields

#### GROUP

The name of the group that has access to the rows, if CA-ICMS is installed.

#### USER

The name of the user who has access to the rows.

#### OWNER

The name of the owner of the table to which row-level security is applied.

#### SECURITY-NAME

The security name assigned to the table.

**STATUS**

Displays the status of the row-level security assignment. This field will be one of the following:

- **(I)—Validation Required**
- **(V)—Validated**
- **(E)—Error in Validation**

This field cannot be changed by the user.

**WHERE-CLAUSE**

The selection criteria that define which rows of the table the user is allowed to access.





# Chapter 16: Extended Screens

---

The Automatic System Facility includes the following extended screens, which can be used by programmers and data processing personnel to achieve full utilization of ASF's capabilities:

- Extended Table Definition screen
- Extended Column Definition screen
- Extended Selection Definition screen
- Extended Key Definition screen
- Extended Table Derivation screen

This section contains the following topics:

[Extended Table Definition Screen—XTAB](#) (see page 241)

[Extended Column Definition Screen—XCOL](#) (see page 245)

[Extended Selection Definition Screen—XSEL](#) (see page 250)

[Extended Key Definition Screen—XKEY](#) (see page 251)

[Extended Table Derivation Screen—XDER](#) (see page 254)

## Extended Table Definition Screen—XTAB

### Description

The Extended Table Definition screen allows you to control aspects of the table definition process that are usually handled automatically by the system. It also provides information about a table defined using ASF menus, so that the table can be used in another environment (for example, in a DML program).

The Extended Table Definition screen can be used by programmers and database administrative personnel for the following purposes:

- To assign a table to a particular area in the database
- To review table information supplied by the system
- To change access to stored tables

Use of this screen is optional; the user does not have to access the Extended Table Definition screen in order to define or use a table through ASF.

### Access

You can access the Extended Table Definition screen in one of the following ways:

- **Select the Extended Table Definition function** from the Table Definition screen; you can select the function by pressing [PF4].
- **Enter XTAB in the screen name field** of any other screen.

### Sample Screen

```

                                CA
XTAB  CA - Automatic System Facility      *Extended Table Definition*
DC560501 MODIFY TABLE DEFINITION
Table Name: JOB

  _ PF1 - Help                          _ PF4 - Extended Key Definition
  _ PF2 - Extended Table Derivation     _ PF5 - Row Level Security
  _ PF3 - Extended Selection Definition

Area: IDMSR-AREA2

Display Access.....: YES              Subschema: RU000239
Load Access.....: YES                  Map.....: RM000239   Version:   1
Change Access.....: YES                 Dialog...: RD000239  Version:   1
Erase Access.....: YES

Generate Map/Dialog.....: YES          Table Definition Length:  120

C/IQMS Modify Option....: YES          Date/Time Created.....:mm/dd/yy 10:20:4
C/IQMS Selective Update: YES           Date/Time Last Updated.:mm/dd/yy 10:50:3

Logical Record Name.....: JOB          Process Standard Area Data: YES
    
```

### Fields

#### Table Name

Displays the name of the table specified on the Activity Selection screen.

#### [PF1] - Help

Invokes the Help screen.

#### [PF2] - Extended Table Derivation

Invokes the Extended Table Derivation screen, which allows you to specify a complex table derivation.

**[PF3] - Extended Selection Definition**

Invokes the Extended Selection Definition screen, which allows you to select the data to be included in the table.

**[PF4] - Extended Key Definition**

Invokes the Extended Key Definition screen, which allows you to establish indexes for the table.

**[PF5] - Row Level Security**

Invokes the Row-level Security screen, which allows you to assign a security name and generic column names to your table.

**Area**

Contains the name of the database area in which the table's data is stored. An ASF administrator can assign the table to a particular area by entering the name of a valid area defined to ASF in this field.

**Display Access**

Controls the generation of an OBTAIN logical-record path in the ASF-generated subschema. YES is the default.

**Load Access**

Controls the generation of a STORE logical-record path in the ASF-generated subschema. YES is the default for stored tables; for views, this field is always NO and cannot be changed.

**Change Access**

Controls the generation of a MODIFY logical-record path in the ASF-generated subschema. YES is the default.

**Erase Access**

Controls the generation of an ERASE logical-record path in the ASF-generated subschema. YES is the default for stored tables; for views, this field is always NO and cannot be changed.

**Generate Map/Dialog**

Indicates whether the system automatically generates a screen format (map) and processing logic (dialog) for accessing the table. The default is YES. When the table is generated, you can access it online through ASF, using this map and dialog. Enter NO if you do not want ASF to generate an online application for using the table.

### C/ICMS Modify Option

Displays the status of the table as it relates to the SAVE UPDATES option of supported PC software. When you request SAVE UPDATE from the PC, ASF acts as follows:

- If the row exists on the mainframe, the request is processed as a MODIFY by ASF.
- If the row doesn't exist on the mainframe, the request is processed as an ADD by ASF.

The valid values for this field are:

- **YES**—If the row doesn't exist on the mainframe, and the request came as a MODIFY from the PC, ASF is allowed to ADD the row.
- **REJECT**—If the row doesn't exist on the mainframe, and the request came as a MODIFY from the PC, ASF is not allowed to ADD the row.

### C/ICMS Selective Update

Enter the status of the table as it relates to the selective update feature applicable to the supported PC software. When a table is to be used with the selective update feature, ASF generates and adds special code to the logical record. This code is generated automatically if the table has only one unique key. The possible values for this field are:

- **YES**—The table has only one unique key and therefore is eligible for selective update. ASF has generated the additional logical record code required to support a selective update.
- **NO**—Selective updates for the table are not allowed, and ASF is not to generate the additional logical record path code.
- **INELIGIBLE**—The table has more than one unique key, or it does not have a unique key; ASF will not generate the additional code.
- **DISABLED**—The table is eligible for selective update, and ASF has generated the additional code. You can change the field to this value and regenerate the table to temporarily suspend selective update activities. The modification of the field in this way does not cause a regeneration of the subschema, it only causes ASF to disallow selective update requests. To return to normal operations, change the field back to YES and regenerate the table.

### Logical Record Name

Displays the name of the logical record defined for a stored table or view in the subschema.

### Subschema

Displays the name of the subschema generated for the table. Subschema names are RUnnnnnn, where *nnnnnn* is the table definition number.

**Map**

Displays the name of the map generated for the table. Map names are RMnnnnnn, where nnnnnn is the table definition number.

**Version**

Displays the map version number.

**Dialog**

Displays the name of the dialog generated for the table. Dialog names are RDnnnnnn, where nnnnnn is the table definition number.

**Version**

Displays the dialog version number.

**Table Definition Length**

Displays the current length of the underlying logical record.

**Date/Time Created**

Shows the date and time the table definition was generated.

**Date/Time Last Updated**

Shows the date and time the table definition was last updated.

**Process Standard Area Data**

Indicates whether ASF will restructure the table when it regenerates the table and whether ASF will delete the table when it deletes the table definition. The default is YES. This field will only appear to those who have administrator authority:

- When a table is deleted
- When a table is regenerated

Enter NO if you do not want ASF to modify the table data when a table definition is regenerated or deleted.

## Extended Column Definition Screen—XCOL

**Description**

The Extended Column Definition screen allows you to adjust the column storage and display criteria that are supplied automatically by the system. The Extended Column Definition screen displays information about one column in the table at a time. You can access the Extended Column Definition screen for selected columns or for all the columns in the table.

### Access

The Extended Column Definition screen *can only be accessed from the Column Definition screen*. On the Column Definition screen, you can select columns for which you want to display extended definitions. To select individual columns, enter any character in the **Ext Screen** field on the Column Definition screen. If you access the Extended Column Definition screen without selecting individual columns, all columns in the table are selected automatically.

You can access the Extended Column Definition screen from the Column Definition screen in one of the following ways:

- **Select the Extended Column Definition function;** you can select the function by pressing [PF2].
- **Enter XCOL in the screen name field.**

To access the extended definition of the next column that was selected on the Column Definition screen, select the **Next Column** function from the Extended Column Definition screen; you can select the function by pressing [PF2].

You can exit from the Extended Column Definition screen and return to the Column Definition screen in one of the following ways:

- **Display all selected columns.** When all columns selected have been displayed, you are returned to the Column Definition screen.
- **Press [Clear].** You are returned to the Column Definition screen and any columns selected are still marked for selection in the Ext Screen field.

**Sample Screen**

```

XCOL  CA - Automatic System Facility          CA          *Extended Column Definition*
DC560407 MODIFY COLUMN DEFINITION
Table Name: JOBS AND APPLICANTS
      _ PF1 - Help      _ PF2 - Next Column      _ PF3 - Message Screen

      Displayed Column Name          Stored Column Name
      SALARY DIFFERENTIAL          SALARY-DIFFERENTIAL

      Display Format          Stored Format
      -Z(05).9(02)          S9(05)V9(02)

Usage: 3 D/0/1/2/3  Display/Comp/Comp-1/Comp-2/Comp-3

Comments.....:          Column Level: WORK
Initial Value:
Calculation...: column(salary requirement) - salary

Edit Table:          Version:          Code Table:          Version:
  U  U/P  Unprotected/Protected  N  B/R/P/T/G/Y/W/N  Blue/Red/Pink/Turquoise
  r  N  Y/N  Blank When Zero      N  N/R  Normal/Reverse Video
  D  B/D/K  Bright/Display/Dark  N  Y/N  Blink/NoBlink

```

**Fields****Table Name**

Displays the name of the table as specified on the Activity Selection screen.

**[PF1] - Help**

Invokes the Help screen.

**[PF2] - Next Column**

Indicates that you can display the extended column definition for the next column you selected on the Column Definition screen by pressing [PF2].

**[PF3] - Message Screen**

Indicates that additional system messages can be displayed on a separate screen. The message display line will instruct you to press [PF3] when there are system messages to be displayed.

**Displayed Column Name**

Displays the name of the column as specified on the Column Definition screen. To change the column name, replace the existing name.

### Stored Column Name

Displays the system-generated name used to store the column. You can replace the existing name with a new name to conform to preferred naming conventions. Note that this name will be changed by the system to conform to COBOL naming conventions, as follows:

- Special characters and spaces are changed to hyphens: !\*#@#SILLY NAME to ----SILLY-NAME.
- Leading and trailing hyphens are deleted: ---SILLY NAME--- to SILLY-NAME.
- Lowercase letters are translated to uppercase letters: silly name to SILLY-NAME.
- Multiple repeating embedded hyphens are reduced to one hyphen: VERY---SILLY---NAME to VERY-SILLY-NAME.
- Numbers in the first position are preceded by an F: 2 VERY SILLY NAMES to F2-VERY-SILLY-NAMES.

**Note:** You cannot change the stored column name for derived columns. If a stored column name conflicts with an existing record, set, or area name in the schema, or with the schema name itself, or if it duplicates the stored column name of another column in the table, then the system-supplied definition number of the column is added to the end of the stored column name. For example, SILLY NAME and !\*#@#SILLY NAME result in the same stored column name, SILLY-NAME. The stored column name for !\*#@#SILLY NAME would be changed to SILLY-NAME-0004.

### Display Format

Displays the external picture for the column (field). To change the external picture, replace the existing picture with a new picture. Refer to "Specifying Column Formats" in the chapter "Extended Functions of ASF" for valid picture values.

### Stored Format

Displays the internal picture for the column (field). To change the internal picture, replace the existing picture with a new picture. Refer to "Specifying Column Formats" in the chapter "Extended Functions of ASF" for valid picture values.

**Note:** You cannot change the stored format for derived columns.

### Usage

Specifies a column usage. Usage is part of the internal picture definition of a column. For more information about the Usage field, see the chapter "Extended Functions of ASF". To create a usage definition, enter one of the following values in the Usage response field:

- **D** for DISPLAY usage (default for text columns)
- **0** for COMP usage
- **1** for COMP-1 usage



- **2** for COMP-2 usage
- **3** for COMP-3 usage (default for numeric columns)

**Initial Value**

Establishes a default value for a column. Enter a 1- to 32-character value that conforms to IDD DDDL conventions for specifying initial column values. Note that the initial value must be consistent with the column type and length. The default initial value for numeric and currency columns is 0 (zero). For alphanumeric columns, the default is blank.

**Calculation**

Establishes a calculation as the basis for obtaining values for column. A calculation can be defined for work-level columns and element-level columns (see Column Level below), but not for group-level columns. Enter a 1- to 60-character arithmetic expression or a literal that is consistent with the column type and length. Literals must be enclosed in single quotation marks. Arithmetic expressions can refer to other columns in the table, either by name or by the ID number assigned to the column on the Extended Selection Definition screen (*#column-id*). For more information about use of this field, see the chapter "Extended Functions of ASF".

**Note:** If a column used in a calculation is deleted from the table, you must modify the calculation and remove any references to the deleted column.

**Column Level**

Indicates the column level, as follows:

- **ELEMENT**—Indicates that the column is an elementary item that contains no subordinate items. When adding a column to a stored table, the column level defaults to ELEMENT, but you can change it to WORK. Derived columns in views are element-level or group-level columns and cannot be changed.
- **WORK**—Indicates that the column is a work column used to store the results of a calculation. A work column is not stored in the database; the Logical Record Facility performs the calculation. All columns added to views are work columns and cannot be changed.
- **GROUP**—Indicates that the column contains subordinate elements. GROUP is displayed only for tables derived from records. You cannot create a group-level column or change GROUP if it is displayed. You cannot delete a group-level column, but you can exclude the column from the screen display.

**Edit Table**

Associates an edit table with a column or changes a table/column association. Enter the 1- to 8-character name of a table that has already been defined and generated through IDD (use ADD TABLE syntax). Observe online mapping conventions for specifying and using edit tables.

**Version**

Specifies the version number of the table named in the Edit Table field. Enter the 1- to 4-character value, from 1 through 9999, of a valid table version.

**Code Table**

Associates a code table with a column or changes a table/column association. Enter the 1- to 8-character name of a table that has already been defined and generated through IDD.

**Version**

Specifies the version number of the table named in the Code Table field. Enter the 1- to 4-character value, from 1 through 9999, of a valid table version.

**Column attributes**

Define criteria for displaying a column. Observe online mapping conventions for specifying field attributes. Define column attributes as follows:

- **UNPROTECTED/PROTECTED** —Enter U (UNPROTECTED) or P (PROTECTED). Values in unprotected columns can be changed; values in protected columns cannot be changed.
- **BLANK WHEN ZERO** —Enter Y (YES) or N (NO).
- **BRIGHT/DISPLAY/DARK** —Enter B (BRIGHT), D (DISPLAY), or K (DARK). BRIGHT columns are displayed in high intensity; DISPLAY columns are displayed in normal intensity; DARK columns are not displayed.
- **BLUE/RED/PINK/TURQUOISE/GREEN/YELLOW/WHITE /NOCOLOR** —Enter B (BLUE), R (RED), P (PINK), T (TURQUOISE), G (GREEN), Y (YELLOW), W (WHITE), or N (NOCOLOR) to specify the column color.
- **NORMAL/REVERSE VIDEO** —Enter N (NORMAL) or R (REVERSE).
- **BLINK/NOBLINK** —Enter Y (BLINK) or N (NOBLINK) to specify either a blinking or a constant column display.

## Extended Selection Definition Screen—XSEL

**Description**

The Extended Selection Definition screen allows you to specify, as part of a table definition, the data to be included in the table. Selection definition is most often used when defining views; you can select the data from the derivation source(s) that you want included in the view.

**Access**

You can access the Extended Selection Definition screen in one of the following ways:

- **Select the Extended Selection Definition function** from the Column Definition screen or from the Extended Table Definition screen; you can select the function by pressing [PF3].
- **Enter XSEL in the screen name field** of any screen.

**Sample Screen**

```

XSEL  CA - Automatic System Facility          CA
DC560603 ENTER SELECTION CRITERIA          *Extended Selection Definition*
Table Name: EMPLOYEE1
      _ PF1 - Help      _ PF2 - Message Screen      Page: 1 of 1

  ID      Column Name      Selection Criteria
0001 EMP-ID-0415
0002 EMP-FIRST-NAME-0415
0003 EMP-LAST-NAME-0415
0004 EMP-STREET-0415
0005 EMP-CITY-0415
0006 EMP-STATE-0415
0007 EMP-ZIP-0415      MATCHES '02###'
0008 EMP-PHONE-0415      MATCHES '617#####'
0009 EMP-STATUS-0415      '01'
0010 EMP-START-YEAR-0415      GE '1980'

Additional Selection Criteria
$7 OR $8

```

**Fields**

The fields on the Extended Selection Definition screen are the same as those on the Data Selection screen. Refer to "Data Selection Screen" in the chapter "Basic Screens" for field descriptions.

## Extended Key Definition Screen—XKEY

**Description**

The Extended Key Definition screen allows you to establish indexes on a table to sort rows for storage or retrieval. Additionally, indexes can be established to increase access efficiency. Note that this screen displays information on any unique key defined on the Column Definition screen. You can override the key information supplied from the Column Definition screen.

The Extended Key Definition screen is intended to be used by personnel with a clear understanding of how data is to be accessed. Refer to *CA IDMS Database Administration Guide* for a detailed discussion of indexing in the CA IDMS/DB environment.

**Access**

You can access the Extended Key Definition screen in one of the following ways:

- **Select the Extended Key Definition function** from the Column Definition screen or from the Extended Table Definition screen; you can select the function from either screen by pressing [PF4].
- **Enter XKEY in the screen name field** of any screen.

**Sample Screen**

```

XKEY  CA - Automatic System Facility          CA          *Extended Key Definition*
DC560906 ADD/MODIFY KEYS
Table Name: JOB
Storage Sequence: 00001 Next/Prior/First/Last/Key Number
Display Sequence: 00001 Same/Key Number
Fast Duplicates: NO 01 02 03 04 05 06
Key Number: 01 02 03 04 05 06
Access Ascending.: YES YES YES YES YES YE
Additional Keys
S
S
Column Name      Keys      Column Order
JOB NUMBER       -          1
DEPARTMENT       -          2
TITLE            x
SALARY           -          1
CURRENT STATUS (FILLED OR OPEN) x
    
```

**Fields**

**Table Name**

Displays the name of the table specified on the Activity Selection screen.

**[PF1] - Help**

Invokes the Help screen.

**Page: \_\_ Of \_\_**

Displays the current screen page number and the total number of pages in the page set. You can move to a particular page by specifying its page number, or page forward using [PF8] and backward using [PF7].

**Storage Sequence**

Specifies the criteria used to store rows. The default is NEXT if no key is defined. If a unique key was specified on the Column Definition screen, the default is the key number assigned to the unique key. You can enter NEXT, PRIOR, FIRST, LAST, or a key number to indicate the order in which rows are to be stored. Specifying a key number stores rows sorted by key, where the key number is a number between 1 and 6 that corresponds to a key in the Additional Keys section of the screen, discussed below.

**Display Sequence**

Specifies the sort criteria used to display rows of a table. The default, SAME, displays data in the same order in which it is stored. You can enter a key number in this field to display data according to a defined sort key that differs from the stored sequence.

**Column Name**

Displays the columns defined for the table.

**Fast Access Keys**

Creates an index on the column. Enter any character next to the desired key column. Any number of columns in a table can be designated as fast access keys. Each fast access key column is indexed in ascending order, duplicates allowed, stored last.

**Additional Keys**

Provides fields in which index keys are specified for the table. Up to six additional keys can be defined for the table. The additional keys can be defined as unique or as allowing duplicates, and can be sorted in ascending or descending order. Use the Duplicates, Ascending, and Column Order fields described below to define each additional key.

**Key Number**

Contains the numbers 01 through 06, which identify the additional key being defined. This screen allows you to define six index keys for the table. The key number is used with the Storage Sequence and Display Sequence fields above to designate which index key is to be used for storing and displaying data.

**Duplicates**

Specifies whether duplicates are allowed. Enter one of the following values to indicate the duplicates option:

- **LAS** (default)—Duplicates are allowed, stored last.
- **FIR**—Duplicates are allowed, stored first.
- **NO**— Duplicates are not allowed.

**Ascending**

Specifies the index order:

- **YES** (default)—Order is ascending.
- **NO**— Order is descending.

### Column Order

Specifies which columns are included in the index key. To designate a single column as the key, enter any numeric value in the Column Order field for the selected column name. To designate a key column made up of multiple columns, enter numbers in their response fields to indicate the order in which to concatenate the columns.

## Extended Table Derivation Screen—XDER

### Description

The Extended Table Derivation screen allows you to perform derivation functions beyond the simple derivation possible on the Table Definition screen. Using the Extended Table Derivation screen, you can:

- Derive a view from more than two data tables
- Derive a view from one or more records

The upper portion of the screen, which is used to identify the source tables, can be paged. Two source tables can be identified on each page; up to six sources can be used to derive a view.

The lower portion of the screen is used to specify join information. Every source table identified must be joined to at least one other source. The lower portion of the screen is not pageable; a maximum of five join operations can be specified.

### Access

You can access the Extended Table Derivation screen in one of the following ways:

- **Select the Extended Table Derivation function** from the Extended Table Definition screen; you can select the function by pressing [PF2].
- **Enter XDER in the screen name field** of any other screen.

**Sample Screen**

XDER CA - Automatic System Facility		CA	*Extended Table Derivation*	
DC560802 ADD/MODIFY DERIVATIONS				
Table Name: EMPLOYEES AND DEPARTMENTS				
			_ PF1 - Help Page: 1 of 1	
		Derivation Source 01		
Source Name.:	EMPLOYEE			
Source Owner:				
Subschema. . .:	EMPSS01	Schema: IDMSR	Version: 1	
		Derivation Source 02		
Source Name.:	DEPARTMENT			
Source Owner:				
Subschema. . .:	EMPSS01	Schema: IDMSR	Version: 1	
	Column #1/Set	Of Der	Oper	Column #2
	DEPT-EMPLOYEE	02	ST	Of Der
				01

**Fields**

**Table Name**

Displays the name of the table as specified on the Activity Selection screen.

**[PF1] - Help**

Invokes the Help screen.

**Page: \_\_ Of \_\_**

Displays the current screen page number and the total number of pages in the page set. Only the top portion of the screen can be paged. You can move to a particular page by specifying its page number, or page forward using [PF8] and backward using [PF7].

**Derivation Source**

Numbers each source table identified. The derivation source numbers are used to specify join information. Two source tables can be identified on each page. ASF numbers each source when it accepts the derivation information.

**Source Name**

Specifies the name of a table, either a stored data table or a non-SQL defined record.

**Source Owner**

Enter the name of the table owner if the source table is a stored data table. You do not need to complete this field if you own the source table.

**Subschema**

If the source table is a non-SQL defined record, enter the name of the subschema in which the record is defined. The subschema must be generated from the system-supplied schema, IDMSR. The source record must be defined in the system-supplied schema, IDMSR.

**Schema**

Displays the schema name IDMSR when a non-SQL defined record has been identified as the source table.

**Version**

Displays the version number of the schema (1).

**Note:** If the source table is a stored data table, complete the Source Name and Source Owner fields under each Derivation Source heading. If the source table is a non-SQL defined record, complete the Source Name and Subschema fields. The Schema and Version fields will be completed automatically.

**Join Information**

When all the source tables have been identified, use the following fields to specify join information. Five join conditions can be specified; each source table identified must participate in at least one join operation.

**Column #1/Set**

Enter the name of the column or set used in the join operation, as follows:

- If you are joining two stored data tables, enter the name of a column in one table that is related to a column in the other table.
- If you are joining two non-SQL defined records, enter the name of the set that connects the two source tables.

**Of Der**

Enter the Derivation Source number supplied for the source table, as follows:

- If you are joining two stored data tables, enter the number of the source table to which the column entered in the Column#1/Set field belongs.
- If you are joining two non-SQL defined records, enter the number of the **owner** of the set specified in the Column#1/Set field.



**Oper**

Enter the operator that defines the condition for joining the two source tables, as follows:

- If you are joining two stored data tables, you will typically enter **EQ** in this field to join Column #1 and Column #2 on the basis of equal values. The following operators can also be specified to join two source records:
  - **NE** specifies that the sources are to be joined when Column #1 is not equal to Column #2.
  - **GT** specifies that the sources are to be joined when Column #1 is greater than Column #2.
  - **LT** specifies that the sources are to be joined when Column #1 is less than Column #2.
  - **GE** specifies that the sources are to be joined when Column #1 is greater than or equal to Column #2.
  - **LE** specifies that the sources are to be joined when Column #1 is less than or equal to Column #2.
- If you are joining two non-SQL defined records, enter **ST** to specify that source table #1 is the owner of source table #2. ST is valid only for non-SQL defined records joined on the basis of a predefined set relationship. The ST operator indicates that the entry in the Column #1/Set field is a set name.

**Column #2**

Enter the join column in another source table that is related to Column #1. Note that you do not specify Column #2 when joining tables on a set relationship.

**Of Der**

Enter the Derivation Source number of the source table to which the column entered as Column #2 belongs. If you are joining tables based on a set relationship, enter the number of the **member** of the set.



# Chapter 17: Administrative Screens

---

This section contains the following topics:

[Administrator Control Screen—CNTL](#) (see page 259)

[Catalog Control Screen—CTLG](#) (see page 262)

[User Limits and Defaults Screen—ULIM](#) (see page 263)

[User Maintenance Screen—USER](#) (see page 267)

[System Table Maintenance Screen—SYST](#) (see page 268)

## Administrator Control Screen—CNTL

### Description

The Administrator Control screen allows you to perform administrative functions for ASF. The Administrator Control screen is available only to users with administrative authority.

### Access

If you have administrative authority, you can access the Administrator Control screen in one of the following ways:

- **Select the Administrator function** from the Activity Selection screen; you can select the function by pressing [PF15].
- **Enter CNTL in the screen name field** of any other screen.

**Sample Screen**

```

                                CA
CNTL  CA - Automatic System Facility      * Administrator Control *
DC560005 MODIFY AND/OR SELECT NEXT ACTIVITY

    _ PF1 - Help                          _ PF5 - Catalog Services
    _ PF2 - Add Secondary Area            _ PF6 - User Limits And Defaults
    _ PF3 - Initialize Definition Area    _ PF7 - User Maintenance
    _ PF4 - Unlock Table Definition       _ PF8 - System Table Maintenance

Unlock Table Definition Number:

Secondary Area Name.....:

                System Defaults

Default Area Name.....: IDMSR-AREA2

Model Map.....: ASF@MAP      Version: 1
Model Dialog...: ASFDGMOD    Version: 1
Model Subschema: ASFSSMOD    Version: 1
    
```

**Fields**

**[PF1] - Help**

Invokes the Help screen.

**[PF2] - Add Secondary Area**

Adds the secondary area named in the Secondary Area Name field (described below). When added, the secondary area is available to ASF for storing data tables.

**[PF3] - Initialize Definition Area**

Initializes the definition area. Perform this function to bring up a new ASF system. This function loads the definition records into the definition area and establishes initial system defaults.

**[PF4] - Unlock Table Definition**

Removes the lock on the table whose definition number was entered in the Unlock Table Definition Number field.

**[PF5] - Catalog Services**

Invokes the Catalog Control screen.

**[PF6] - User Limits And Defaults**

Invokes the User Limits and Defaults screen.

**[PF7] - User Maintenance**

Invokes the User Maintenance screen.

**[PF8] - System Table Maintenance**

Invokes the System Table Maintenance screen.

**Unlock Table Definition Number**

To unlock a table definition, enter the definition number assigned to the locked table and press [PF4].

**Secondary Area Name**

Enter the name of the secondary area you are adding. This field is used in conjunction with the Add Secondary Area function. The secondary area named must already be prepared according to the procedure for adding database areas outlined in *CA IDMS Database Administration Guide*. Once added, the area can be used by ASF to store data tables.

**System Defaults**

The following fields are used to specify **System Defaults** to be used by ASF:

**Default Area Name**

Specify the system default area for storing data tables. It can be overridden for a user or a table by using the User Limits and Defaults screen or the Extended Table Definition screen.

**Model Map**

To replace the model map used to generate all ASF screens, enter the replacement map name, as defined in the data dictionary. Use this option to change the map name or literals to meet site standards.

**Version**

Specify the version number of the replacement model map.

**Model Dialog**

To override the default model used to generate the ASF dialog, enter the new name of the model dialog, as defined in the data dictionary. Use this option to change *only* the name to meet site standards.

**Version**

Enter the version number of the new model dialog.

**Model Subschema**

To override the default module containing model subschema syntax used to generate the subschema for a table, enter the new name of the model subschema, as defined in the data dictionary. Use this option to change *only* the name to meet site standards.

**Version**

Enter the version number of the new model subschema.

## Catalog Control Screen—CTLG

**Description**

The Catalog Control screen allows you to perform special administrative functions for the catalog. This screen is available only to users with administrative authority.

**Access**

If you have administrative authority, you can access the Catalog Control screen in one of the following ways:

- **Select the Catalog Services function** from the Administrator Control screen; you can select the function by pressing [PF5].
- **Enter CTLG in the screen name field** of any screen.

**Sample Screen**

```
CTLG  CA - Automatic System Facility      CA      *** Catalog Control ***
DC422007 SELECT AN ACTION

                                           _ PF1 - Help

Select Desired Option:

  _ Lock User
  _ Unlock User
  _ Delete All Object Entities In The Catalog
  _ Give Administrator Authority
  _ Remove Administrator Authority
  _ Assign User Limits And Defaults

User Name:
```

**Fields****[PF1] - Help**

Invokes the Help screen.

**Lock User**

Enter any character to prevent the named user from accessing ASF.

**Unlock User**

Enter any character to allow a previously locked user to access ASF.

**Delete All Object Entities In The Catalog**

Enter any character to delete all object entries from the catalog. When the area is initialized, all object entries must be deleted from the catalog to ensure compatibility between catalog entries and the data. This allows you to retain your catalog structure (other than objects) because you do not have to reinitialize the dictionary. You will be prompted to press [Enter] before the objects are deleted. *Use this option with extreme caution — refer to CA IDMS Database Administration Guide for further details.*

**Give Administrator Authority**

Enter any character to give the named user administrative authority, which allows the user to use the administrator screens.

**Remove Administrator Authority**

Enter any character to remove administrator authority from the named user.

**Assign User Limits And Defaults**

Enter any character to access the User Limits and Defaults screen. The User Limits and Defaults screen allows you to override the system default space allocation for the user, override the model defaults, and restrict passkeys.

**User Name**

Enter the name of the user to be processed. The user name must be specified to lock or unlock a user, or to give or remove administrator authority.

## User Limits and Defaults Screen—ULIM

**Description**

The User Limits and Defaults screen allows you to perform the following functions:

- Assign space allocations for the user that override the system defaults
- Display the amount of space currently available for storing the user's tables

- Assign an area for storing the user's tables
- Assign the default model map, dialog, and subschema for the user
- Assign default passkey authorization; for example, giving only LIST and COPY passkeys to a user allows the user to access data but not to create or modify data

The changes made on the User Limits and Defaults screen do not affect the user's current ASF session. These changes take effect either the next time the user signs on to ASF or when the user leaves ASF and gets back in. This screen is available only to users with administrative authority.

**Access**

If you have administrative authority, you can access the User Limits and Defaults screen in one of the following ways:

- **Select the User Limits And Defaults function** from the Administrator Control screen; you can select the function by pressing [PF6].
- **Select the Assign User Limits And Defaults function** from the Catalog Control screen.
- **Enter ULIM in the screen name field** of any screen.

**Sample Screen**

```

                                CA
ULIM  CA - Automatic System Facility      * User Limits And Defaults *
DC422005 ENTER USER ID

User Name:                                _ PF1 - Help

Allocation Limits And Defaults For User's Tables      Unstructured
                                                    Objects
                                                    (IN K BYTES)
                (NO. OF TABLES)
Total Allocation Limit.....:

Current Total Allocation...:

Area Name.....:

Default Models For Tables

Map.....:          Version:
Dialog...:         Version:
Subschema:         Version:

Passkey Authorization (Specify Y Or N)

Add.....:          Copy....:          Create:          Modify:
Erase....:        Redefine:         List..:
    
```



**Fields****User Name**

Enter the name of the user to whom the defaults apply. When you press [Enter], ASF will either display the existing user limits and defaults or return a USER DEFAULTS HAVE NOT BEEN ASSIGNED message. Now you can proceed to assign (or reassign) user limits and defaults.

**[PF1] - Help**

Invokes the Help screen.

**Allocation Limits And Defaults For User's Tables**

Establish the amount of storage allocated for the user's tables and unstructured objects, as follows:

**Total Allocation Limit (No. of Tables)**

The first column displays the total number of tables allowed for the user. The second column (Unstructured Objects) displays the total number of bytes of storage allowed for all unstructured objects owned by the user.

**Current Total Allocation**

The first column displays the total number of tables allocated for the user. The second column displays the total number of bytes currently allocated for the user's unstructured objects.

**Area Name**

Enter the name of the area that will hold tables owned by the user. This area name will override the system default area and can be overridden on the Extended Table Definition screen by an ASF administrator.

**Default Models For Tables**

Allows you to override system default model names for the user:

**Map**

Enter the name of the default model map. This model map will override the system default map name.

**Version**

Enter the version number of the default model map.

**Dialog**

Enter the name of the module containing model dialog syntax to override the system default model module.

**Version**

Enter the version number of the default model dialog.

**Subschema**

Enter the name of the module containing model subschema syntax to override the system default model module.

**Version**

Enter the version number of the default model subschema.

**Passkey Authorization (Specify Y Or N)**

Establish authorizations for the user that override any passkeys held through ownership or assignment. For example, disallowing Create access to DATA-ENTRY-CLERK means that the user can add rows to an existing table, but cannot define a new table.

**Add**

Enter Y or N to allow or disallow Add access, that is, to prevent the user from adding rows to existing tables.

**Erase**

Enter Y or N to allow or disallow Erase access.

**Copy**

Enter Y or N to allow or disallow Copy access.

**Redefine**

Enter Y or N to allow or disallow Redefine access, that is, to prevent the user from altering the definition of an existing table.

**Create**

Enter Y or N to allow or disallow Create access.

**List**

Enter Y or N to allow or disallow List access.

**Modify**

Enter Y or N to allow or disallow Modify access, that is, to prevent the user from altering existing data in tables.

## User Maintenance Screen—USER

### Description

The User Maintenance screen allows the ASF administrator to add a user to the catalog or delete a user from the catalog. Users must be entered in the catalog in order to be assigned limits and defaults or to be given administrative authority.

Users are automatically entered in the catalog when they sign on to ASF for the first time. The User Maintenance screen allows the ASF administrator to assign limits and defaults to the user before the user accesses ASF for the first time.

To add a user to the catalog, the user must be defined in the data dictionary. When added, the user will automatically be given access to ASF in the data dictionary if access to ASF was not previously defined.

The ASF administrator can delete users from the catalog to revoke their access to ASF. To delete a user, the user's directory must be empty (the user cannot own any tables). The user's dictionary definition is automatically modified; the ACCESS TO ASF clause is deleted.

### Access

If you have administrative authority, you can access the User Maintenance screen in one of the following ways:

- **Select the User Maintenance function** from the Administrator Control screen; you can select the function by pressing [PF7].
- **Enter USER in the screen name field** of any screen.

### Sample Screen

```
USER  CA - Automatic System Facility          CA          ** User Maintenance **
DC560920 ENTER USER NAME

                                           _ PF1 - Help

User Name:

Select Action To Be Performed:
    _ Add User
    _ Delete User
```

### Fields

#### [PF1] - Help

Invokes the Help screen.

#### User Name

Enter the name of the user to be added or deleted.

#### Add User

Enter any character to add the named user to the catalog. The named user must already be defined in the data dictionary; the user's dictionary definition is modified automatically, giving the user access to ASF.

#### Delete User

Enter any character to delete the named user from the catalog. The user cannot own any tables. The user's data dictionary definition is modified automatically, removing the user's access to ASF.

## System Table Maintenance Screen—SYST

### Description

The System Table Maintenance screen allows the DBA (or anyone with DBA authority) to access the following tables directly:

- \$UNSTRUCTURED-IDB-OBJECT\$
- \$SECURITY-RUNTIME-TABLE\$

- \$OBJECT-SECURITY-NAME-TABLE\$
- \$SRT-OST-CROSS-REFERENCE\$

### Access

You can access the System Table Maintenance screen in one of the following ways:

- **Select the System Table Maintenance function** from the Administrator Control screen; you can select the function by pressing [PF8].
- **Enter SYST in the screen name field** of any screen.

You can leave the System Table Maintenance screen in one of the following ways:

- **Select an activity from the menu and press [Enter]**; the appropriate screen for performing the activity you select will be displayed.
- **Press [Clear]** to return to the prior screen.
- **Enter the screen name of another screen** in the screen name field.

### Sample Screen

```

SYST  CA - Automatic System Facility          CA          *System Table Maintenance*
DC560727 SELECT A SYSTEM TABLE

Table Name:

      _ PF1 - Help                               _ PF4 - Display/Change Data
      _ PF2 - Define Table                       _ PF5 - Select Data
      _ PF3 - Load Data                         _ PF6 - Query

      _ $UNSTRUCTURED-IDB-OBJECT$
      _ $SECURITY-RUNTIME-TABLE$
      _ $OBJECT-SECURITY-NAME-TABLE$
      _ $SRT-OST-CROSS-REFERENCE$

```

### Fields

#### Table Name

Displays the name of the table as specified on the System Table Maintenance screen.

#### [PF1] - Help

Invokes the Help screen.

**[PF2] - Define Table**

Invokes the Table Definition screen, which enables you to define a new table or to modify or delete an existing table.

**[PF3] - Load Data**

Invokes the Data Display screen for an existing table, which enables you to load data to be stored in the table.

**[PF4] - Display/Change Data**

Invokes the Data Display screen for an existing table, which enables you to display, change, delete, and load data.

**[PF5] - Select Data**

Invokes the Data Selection screen for a generated table, which enables you to select data for display by specifying selection criteria.

**[PF6] - Query**

Invokes OnLine Query, which enables you to display a generated table in row and column format. This function applies only if OnLine Query is installed at your site.

# Chapter 18: IDMSRADM

---

This section contains the following topics:

- [Description](#) (see page 271)
- [Syntax](#) (see page 271)
- [Input Parameter Statements](#) (see page 273)
- [General Usage](#) (see page 286)
- [JCL](#) (see page 287)

## Description

### Purpose

Using the relational database administration utility, IDMSRADM, you can:

- Monitor space utilization by user
- Import tables to specific areas
- Export table definitions to a sequential file with or without the table data
- Delete tables and (if CA-ICMS is installed) post office objects by owner and path name

These features let you manage and delete objects more specifically.

## Syntax

### SIGNON Statement

```
▶▶ SIGNon user-name [ PASsword password ] [ DBName dictionary-name ] [ NODename node-name ] . ▶▶
```

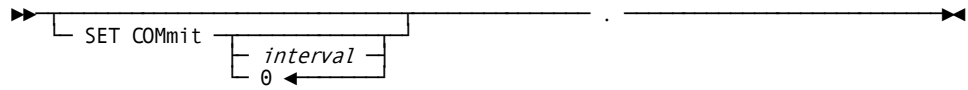
### SET AREA Statement

```
▶▶ SET AREa [ area-name ] . ▶▶
```

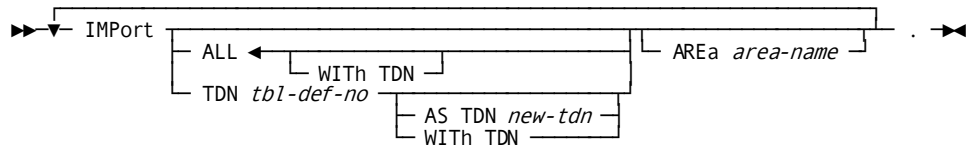
### SET OWNER Statement

```
▶▶ SET OwNer [ owner-name ] . ▶▶
```

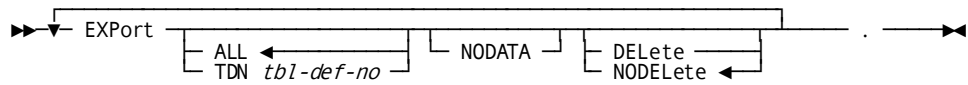
**SET COMMIT Statement**



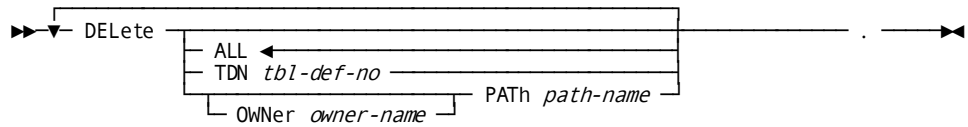
**IMPORT Statement**



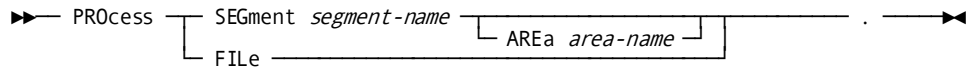
**EXPORT Statement**



**DELETE Statement**



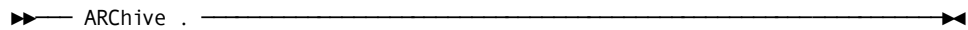
**PROCESS Statement**



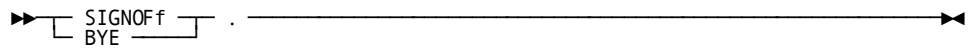
**REPORT Statement**



**ARCHIVE Statement**



**SIGNOFF/BYE Statement**





## Input Parameter Statements

### Parameter Statement Descriptions

IDMSRADM utility processing is controlled by input parameter statements:

Statement	Description
SIGNON	Initiates IDMSRADM processing
SET AREA	Optionally specifies the default area into which to import tables
SET OWNER	Optionally specifies the default owner from which to import or delete tables
SET COMMIT	Optionally specifies the frequency at which IDMSRADM issues COMMIT commands while importing or deleting tables
IMPORT	<ul style="list-style-type: none"> <li>■ Specifies table(s) to import</li> <li>■ Can temporarily override SET AREA</li> </ul>
EXPORT	Specifies table(s) to export
DELETE	<ul style="list-style-type: none"> <li>■ Specifies table(s) to delete</li> <li>■ Can temporarily override SET OWNER</li> </ul>
PROCESS	Specifies the area or file to be processed for REPORT.
REPORT	Initiates the processing of the Space Utilization report.
ARCHIVE	Exports all objects in the CA-ICMS archive directory
SIGNOFF/BYE	Terminates IDMSRADM processing

## SIGNON Statement

### Purpose

The SIGNON statement initiates IDMSRADM processing. The statement specifies the name and password of a user and, optionally, identifies the dictionary that describes the database containing the data tables to be exported or imported.

One SIGNON statement is required for each IDMSRADM run.

### Syntax

```

▶▶▶ SIGNon user-name [ PASsword password ] [ DBName dictionary-name ] [ NODename node-name ] . ▶▶▶

```

### Parameters

#### *user-name*

The name of a user who has authority to access ASF.

#### PASsword *password*

The password of the named user.

A password that contains embedded blanks must be enclosed in quotation marks.

The password must be specified if the named user is associated with a password in the data dictionary.

#### DBName *dictionary-name*

The name of the data dictionary containing the catalog associated with the data tables to be exported or imported.

By default, if you do not specify a dictionary name, IDMSRADM uses the system default (primary) dictionary.

Code the DBNAME parameter only in a multiple dictionary environment.

#### NODename *node-name*

DDS only.

The DDS node that controls the data dictionary to be accessed by IDMSRADM.

### Usage

#### z/OS Systems

The DDS node name (NODENAME) and the data dictionary name (DBNAME) can be specified in //SYSIDMS DD \*.

## SET AREA Statement

### Purpose

The SET AREA statement assigns a default area to which tables are imported.

### Syntax

```
▶▶ SET AREa [ area-name ] . ▶▶
```

### Parameters

#### *area-name*

The name of an area to which tables will be imported.

""

Specifies that the area to which data tables are imported is the same area from which they were exported.

### Usage

You can override the default target area by specifying an area name in the IMPORT statement.

## SET OWNER Statement

### Purpose

The SET OWNER statement assigns a default owner to tables that are imported or deleted.

### Syntax

```
▶▶ SET OWNeR [ owner-name ] . ▶▶
```

**Parameters**

***owner-name***

The ID of a user defined to either ASF or CA-ICMS.

" "

Specifies the current owner of the tables to be imported or deleted.

**Usage**

You can change the default owner name by:

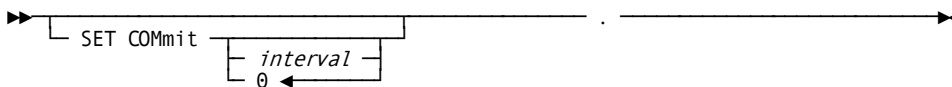
- Repeating the SET OWNER statement before subsequent IMPORT or DELETE statements.
- Specifying a different owner in the DELETE statement.

## SET COMMIT Statement

**Purpose**

The SET COMMIT statement specifies the frequency at which IDMSRADM issues COMMIT commands when a table is imported or deleted.

**Syntax**



**Parameters**

***interval***

The number of STORE (for IMPORT) or ERASE (for DELETE) operations IDMSRADM is to perform before issuing a COMMIT.

By default, if you do not specify an interval, IDMSRADM will not issue a COMMIT until the IMPORT or DELETE operation is complete.

**0**

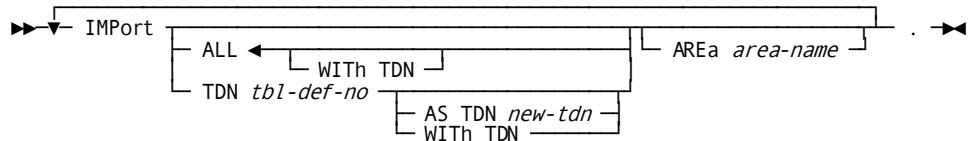
Directs IDMSRADM not to issue a COMMIT until the IMPORT or DELETE operation is complete.

## IMPORT Statement

### Purpose

Copies one table or all tables from a file to an area.

### Syntax



### Parameters

#### ALL

Copies all tables from the file.

By default, if you do not specify a table definition number, each imported table will get the next available table definition number.

#### WITH TDN

Specifies that each imported table get its original table definition number.

#### TDN

Specifies the table to import.

#### *tbl-def-no*

The table definition number of the table to import.

#### AS TDN

Directs IDMSRADM to give the table a new table definition number.

*new-tdn* The new table definition number of the imported table.

#### WITH TDN

Directs IDMSRADM to give the table its original table definition number.

#### AREa *area-name*

The name of the area to import the table(s) into.

### Usage

You can reset the default area with another SET statement.

You can override the default area by specifying another area on an IMPORT statement.

Specifying the area as a blank enclosed by double quotes (" ") sets the import area to the area from which the table was exported.

### Conflicting Table Definition Numbers

If the target area already contains a table with the specified or default table definition number of a table to be imported, IDMSRADM prints an error message and does not import the table.

If you are importing more than one table, and some have conflicting table definition numbers, an error message will be printed for each case of conflict, but tables with non-conflicting table definition numbers will be imported.

### Default Area Names

If you do not specify an area in the IMPORT statement, IDMSRADM imports into the area specified by the SET AREA statement. If you did not include a SET AREA statement, IDMSRADM uses the name of the area from which the table was exported.

### Importing Tables to a Specific Area

You can import either single or multiple tables to a named area.

#### *Import single tables to an area*

To import single tables to an area, specify the area on the IMPORT statement.

#### *Import two or more tables to the same area*

To import two or more tables to the same area, set a default area name prior to the IMPORT statement.

#### *Subsequent imports to the same area*

For subsequent imports to the same area, you don't have to re-specify the area. You can override the default area by specifying an area on another IMPORT statement. You can also change the default by specifying another SET command.

### Examples

#### Example 1

##### *Importing specific tables to a default area*

```
SET AREA ABC
IMPORT TDN 123.
IMPORT TDN 204.
SET AREA " ".
IMPORT TDN 333.
IMPORT TDN 527.
BYE.
```

**Example 2**

*Overriding the default area to import specific tables*

```
SET AREA ABC
IMPORT TDN 123.
IMPORT TDN 572 AREA EMP.
IMPORT TDN 415.
IMPORT TDN 501 AREA EMP.
BYE.
```

**Example 3**

*Specifying a default owner for importing*

```
SET AREA DEPT
SET OWNER USER1.
IMPORT TDN 123.
IMPORT TDN 395.
SET OWNER " ".
IMPORT TDN 934.
BYE.
```

**Example 4**

*Specifying commits*

```
SET COMMIT 25.
SET AREA DEPT.
IMPORT TDN 802.
BYE.
```

**Example 5**

*Importing tables with table-definition numbers*

```
SET AREA EMP.
SET OWNER USER1.
IMPORT TDN 415 WITH TDN.
IMPORT TDN 578 AS TDN 398.
```

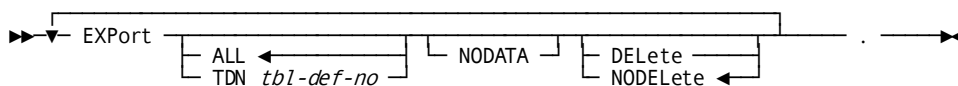
## EXPORT Statement

**Purpose**

Copies one table or all tables from an area to a sequential file.

The table(s) can optionally be deleted from the area.

**Syntax**



**Parameters**

**ALL**

Exports all tables from the area.

**TDN**

Exports a specific table from the area.

***tbl-def-no***

The table definition number of the table to be exported.

**NODATA**

Exports the definition of the table only.

**DElete**

Deletes the table(s) from the area.

By default, if you do not specify DELETE, IDMSRADM does not delete the exported table(s) from the area.

**NODElete**

Directs IDMSRADM not to delete the table(s) from the area.

This is the default.

**Examples**

**Example 1**

*Exporting table definitions only*

```
EXPORT ALL NODATA.
BYE.
```

**Example 2**

*Deleting tables after export*

```
SET OWNER USER1.
EXPORT PATH DEPT.
EXPORT TDN 218 DELETE.
EXPORT TDN 923 NODEL.
BYE.
```

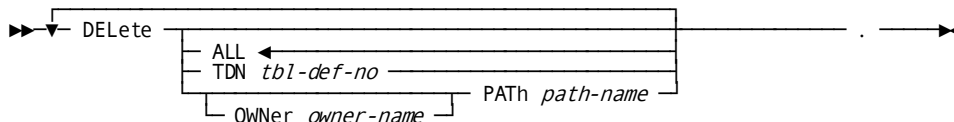


## DELETE Statement

### Purpose

Deletes one or all tables from an area.

### Syntax



### Parameters

#### ALL

Deletes all tables from the area specified in the SET AREA statement.

This is the default.

#### TDN *tbl-def-no*

Specifies a table to delete with its table definition number.

By default, if you do not specify a table to delete, all tables in the area specified in the SET AREA statement are deleted.

#### OWNeR *owner-name*

Specifies a table to delete with the name of the table's owner.

The owner can also be specified with the SET OWNER statement.

By default, if you do not specify a table to delete, all tables in the area specified in the SET AREA statement are deleted.

#### PATH *path-name*

Specifies a table to delete with the path name of the table.

By default, if you do not specify a table to delete, all tables in the area specified in the SET AREA statement are deleted.

*Path-name* consists of either the object name alone, if it is unique, or a string consisting of one or more folder names (if CA-ICMS is installed) and an object name separated by a period to uniquely identify the object.

If a folder name or object name contains a space, period, or single quotation mark, enclose it in double quotation marks (" ").

## Usage

### Deleting Tables by Owner and Path Names

*To delete tables by owner and path*

To delete tables by owner and path, specify the owner name and the path name instead of the table-definition number in the DELETE statement.

*To delete multiple tables for one owner*

To delete multiple tables for one owner, set a default owner name on the SET statement. You don't have to re-specify the owner name to delete subsequent tables for the same owner.

You can override the default by specifying an owner and path name in the DELETE statement. You can also change the default by specifying another SET command.

When specifying a path name, include as many folder names (separated by periods) as it takes to uniquely identify the object.

If the owner or path name contains a space, period, or single quote, then the owner or path name must be specified in double quotes.

### Deleting post office objects

You can delete a post office object by specifying its table definition number.

Alternatively, to delete a post office object by owner and path, assign "POST OFFICE" to *owner-name*, followed by the path name for the post office object:

```
DELETE OWNER "POST OFFICE"  
      PATH path-name
```

### Deleting a Post Office

A post office is deleted automatically when all of its objects are deleted.

## Examples

### Example 1

*Deleting table by default owner names*

```
SET OWNER USER1.  
DELETE PATH ABC.  
SET OWNER USER2..  
DELETE PATH DEF.  
BYE.
```

**Example 2**

*Deleting tables by owner and path names*

```
SET OWNER USER1.
DELETE PATH "FOLDERA.ABC"
SET OWNER USER2.
DELETE PATH "TABLE ONE".
DELETE OWNER GSR PATH EMP.
BYE.
```

**Example 3**

*Specifying commits*

```
SET COMMIT 10.
SET OWNER USER2.
DELETE PATH DEPT.
DELETE TDN 702.
BYE.
```

## PROCESS Statement

**Purpose**

The PROCESS statement specifies the area or file to be processed.

**Syntax**

```
►► PROCESS [ SEGment segment-name [ AREa area-name ] ] [ FILE ] . ►►
```

**Parameters****SEGment *segment-name***

Specifies the name of a specific segment in the DMCL.

**AREa *area-name***

Specifies the name of a specific area in the DMCL.

**FILE**

Specifies the file that contains a previously created report.

**Usage**

*Multiple PROCESS statements*

You can specify multiple areas using multiple PROCESS statements.

### When Gathering Information

If you are gathering information to use for processing subsequent reports, use the PROCESS SEGMENT parameter and specify a new data set name on the SYS004 statement in the JCL.

### Processing Information in a Data Set

If the information you want to process is in a data set, use the PROCESS FILE parameter and specify the name of the data set on the SYS004 statement in the JCL.

## REPORT Statement

### Purpose

The REPORT statement initiates the processing of the Space Utilization report.

### Syntax



### Parameters

#### ALL

Specifies a Space Utilization report containing information on every user. ALL is the default.

#### USER *user-name*

Specifies a Space Utilization report containing information on the name of a specific user previously defined to ASF or CA-ICMS.

#### DETAIL

Generates a report with totals for each table of each user. Totals are listed for each user. DETAIL is the default.

#### SUMMARY

Generates a report of space with totals listed for each user.

### Usage

You can monitor space utilization for a specific user or for all users by producing summary and detail reports. The table below describes each kind of report:

Report type	Description
USER SUMMARY	Contains total of space used for one user

Report type	Description
USER DETAIL	Contains totals of space used for each table of one user
ALL SUMMARY	Contains totals of space used for every user
ALL DETAIL	Contains totals of space used for each table of every user

### How to Generate a Space Utilization Report

Follow the steps below to generate a Space Utilization report:

1. Specify one or all areas, or specify no area if processing from a file
2. Specify the types of reports you want

## ARCHIVE Statement

### Purpose

The ARCHIVE statement exports all objects in the CA-ICMS archive directory.

### Syntax

```
▶▶ ARCHIVE .
```

## SIGNOFF/BYE Statement

### Purpose

The SIGNOFF statement terminates IDMSRADM processing.

One SIGNOFF statement is required for each IDMSRADM run.

SIGNOFF and BYE are synonyms, and can be used interchangeably.

### Syntax

```
▶▶ SIGNOFF .
   BYE
```

## General Usage

### Input

Input to IDMSRADM consists of:

- The database or sequential file containing the objects to be processed
- Statements to control the utility processing

### Output

The IDMSRADM utility generates a message listing that reports on the objects processed.

In addition, depending on the operation performed, IDMSRADM may change the contents of:

- The sequential file containing the processed objects
- The database
- The catalog segment of the dictionary

### Function Exclusivity

IDMSRADM functions are exclusive. You cannot mix IMPORT, EXPORT, DELETE, or REPORT on the same IDMSRADM run.

### Execution Mode

You can execute the IDMSRADM utility in local mode or under central version.

### Sample output

Below is a sample Space Utilization report:

```

mm/dd/yy 11:08:42      IDMSRADM      - CA/IOMS DATABASE ADMINISTRATION UTILITY      PAGE:  1
SIGNON USER "CULL DBA" PASSWORD ??????? DBNAME ASFDICT .
PROCESS SEGMENT ASFDICT .
REPORT ALL DETAIL .
BYE .
mm/dd/yy 11:08:42      IDMSRADM      - CA/IOMS DATABASE ADMINISTRATION UTILITY      PAGE:  2
                                     SPACE UTILIZATION BY USER
                                     DMCL: CVDMLCL      ALL DETAIL

USER NAME      OBJECT  TABLE
OBJECT NAME    TYPE    NUMBER  SEGMENT  AREA      SPACE USED
CULL DBA
DEMO           DATATABLE  107  ASFDICT  IDMSR-AREA2      224
USER LIMIT:    NONE      USER TOTALS:      224
mm/dd/yy 11:08:42      IDMSRADM      - CA/IOMS DATABASE ADMINISTRATION UTILITY      PAGE:  3
                                     SPACE UTILIZATION BY USER
                                     DMCL: CVDMLCL      ALL DETAIL

USER NAME      OBJECT  TABLE
OBJECT NAME    TYPE    NUMBER  SEGMENT  AREA      SPACE USED
JPD
MYTAB         DATATABLE  115  ASFDICT  IDMSR-AREA2      56
USER LIMIT:    NONE      USER TOTALS:      56
mm/dd/yy 11:08:42      IDMSRADM      - CA/IOMS DATABASE ADMINISTRATION UTILITY      PAGE:  4
                                     SPACE UTILIZATION BY USER
                                     DMCL: CVDMLCL      ALL DETAIL

USER NAME      OBJECT  TABLE
OBJECT NAME    TYPE    NUMBER  SEGMENT  AREA      SPACE USED
USER1
DATATYP       DATATABLE  112  ASFDICT  IDMSR-AREA2      312
GENTAB2       DATATABLE  114  ASFDICT  IDMSR-AREA2      56
GENTEST       DATATABLE  111  ASFDICT  IDMSR-AREA2      56
STAMPTST      DATATABLE  113  ASFDICT  IDMSR-AREA2      192
TABLE1        DATATABLE  101  ASFDICT  IDMSR-AREA2      896
TESTTAB-103   DATATABLE  106  ASFDICT  IDMSR-AREA2      1,232
TEST2         DATATABLE  105  ASFDICT  IDMSR-AREA2      388
TEST8         DATATABLE  110  ASFDICT  IDMSR-AREA2      56
USER LIMIT:    300      USER TOTALS:      3,188
mm/dd/yy 11:08:42      IDMSRADM      - CA/IOMS DATABASE ADMINISTRATION UTILITY      PAGE:  5
                                     SPACE UTILIZATION BY USER
                                     DMCL: CVDMLCL      ALL DETAIL

USER NAME      OBJECT  TABLE
OBJECT NAME    TYPE    NUMBER  SEGMENT  AREA      SPACE USED
TOTALS FOR THIS REPORT .....
mm/dd/yy 11:08:42      IDMSRADM      - CA/IOMS DATABASE ADMINISTRATION UTILITY      PAGE:  6
                                     SPACE UTILIZATION BY USER
                                     DMCL: CVDMLCL      ALL DETAIL

*** END OF JOB ***

```



In the sections to follow is JCL for these operating systems:

- z/OS
- z/VSE
- z/VM

## z/OS—Central Version (CV)

### JCL to execute IDMSRADM in CV mode z/OS

```
//RADM      EXEC PGM=IDMSRADM,REGION=1024K
//STEPLIB  DD   DSN=idms.dba.loadlib,DISP=SHR
//          DD   DSN=idms.custom.loadlib,DISP=SHR
//          DD   DSN=idms.cagjload,DISP=SHR
//sysctl   DD   DSN=idms.sysctl,DISP=SHR
//dcmsg    DD   DSN=idms.sysmsg.ddldcmsg,DISP=SHR
//SYS004   DD   DSN=radm.archive,DISP=(NEW,CATLG,DELETE),
//          UNIT=disk,VOL=SER=nnnnnn,SPACE=(TRK,(nn,nn)),
//          DCB=BLKSIZE=5492
//SYSLST   DD   SYSOUT=A
//SYSIDMS  DD   *
DMCL=dmcl-name
```

*put other SYSIDMS parameters, as required, here*

```
/*
//SYSIPT   DD   *
```

*put IDMSRADM parameter statements here*

Item	Description
idms.dba.loadlib	Data set name of the load library containing the DMCL and database name table load modules
idms.custom.loadlib	Data set name of the load library containing customized load modules
idms.cagjload	Data set name of the load library containing the CA IDMS executable modules that do not require customization
idms.sysctl	Data set name of the SYSCTL file
radm.archive	Data set name of the sequential file; can be output to disk
disk	Symbolic device name of the SYS004 file
5492	Block size of the SYS004 file; must be a multiple of 5492

**Note:** The DCB RECFM and LRECL specifications for the sequential file are built into IDMSRADM; the utility will override user specifications for these parameters.



## z/OS—Local Mode

### JCL to execute IDMSRADM in local mode (z/OS)

```
//RADM      EXEC PGM=IDMSRADM,REGION=1024K
//STEPLIB  DD DSN=idms.dba.loadlib,DISP=SHR
//          DD DSN=idms.custom.loadlib,DISP=SHR
//          DD DSN=idms.cagjload,DISP=SHR
//asfdml   DD DSN=idms.asfdict.ddldml,DISP=SHR
//asflod   DD DSN=idms.asfdict.asflod,DISP=SHR
//asfdefn  DD DSN=idms.asfdict.asfdefn,DISP=SHR
//asfdata  DD DSN=idms.asfdict.asfdata,DISP=SHR
//dcmsg    DD DSN=idms.sysmsg.ddldcmsg,DISP=SHR
//sysjrnl  DD DSN=idms.tapejrnl,DISP=???
//SYS004   DD DSN=radm.archive,DISP=(NEW,CATLG,DELETE),
//          UNIT=disk,VOL=SER=nnnnnn,SPACE=(TRK,(nn,nn)),
//          DCB=BLKSIZE=5492
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,(nn,nn))
//SORTWK02 DD UNIT=SYSDA,SPACE=(TRK,(nn,nn))
//SORTMSG  DD SYSOUT=A
//SYSLST   DD SYSOUT=A
//SYSIDMS  DD *
DMCL=dmcl-name
```

*put other SYSIDMS parameters, as required, here*

```
/*
//SYSIPT   DD *
```

*put IDMSRADM parameter statements here*

Item	Description
idms.dba.loadlib	Data set name of the load library containing the DMCL and database name table load modules
idms.custom.loadlib	Data set name of the load library containing the CA IDMS customized executable modules
idms.cagjload	Data set name of the load library containing the CA IDMS executable modules that do not require customization
asfdml	Ddname of the asf dictionary definition (DDLDMML) area
idms.asfdict.ddldml	Data set name of the asf dictionary definition (DDLDMML) area
asflod	Ddname of the asf dictionary definition load (DDLDCLOD) area
idms.asfdict.asflod	Data set name of the asf dictionary definition load (DDLDCLOD) area

Item	Description
asfdefn	Ddname of the asf data definition (IDMSR-AREA) area
idms.asfdict.asfdefn	Data set name of the asf data definition area (IDMSR-AREA) area
asfdata	Ddname of the asf data (IDMSR-AREA2) area
idms.asfdict.asfdata	Data set name of the asf data area (IDMSR-AREA2) area
idms.sysctl	Data set name of the SYSCTL file
radm.archive	Data set name of the sequential file; can be output to disk
disk	Symbolic device name of the SYS004 file
5492	Block size of the SYS004 file; must be a multiple of 5492

**Note:** The DCB RECFM and LRECL specifications for the sequential file are built into IDMSRADM; the utility will override user specifications for these parameters.

## z/VSE—Local Mode

**Note:** IDMSRADM may need to be relinked with an IDMSDBLP module before being executed.

For information on the IDMSDBLP module, see the z/VSE JCL chapter in *CA IDMS Utilities Guide*.

### JCL to execute IDMSRADM in local mode (z/VSE)

```
// EXEC PROC=IDMSLBLS
// DLBL radmarc, 'radm.archive', 0
// EXTENT SYS004, nnnnnn, 1, , ssss, tttt
// ASSGN SYS004, DISK, VOL=nnnnnn, SHR
// DLBL SORTWK1, 'sort.work1', 0
// EXTENT sysnnn, nnnnnn, 1, , ssss, tttt
```

```
// ASSGN sysnnn,DISK,VOL=nnnnnn,SHR
// DLBL SORTWK2,'sort.work2',0
// EXTENT sysnnn,nnnnnn,1,,ssss,tttt
// ASSGN sysnnn,DISK,VOL=nnnnnn,SHR
// LIBDEF *,SEARCH=CA IDMS nn.n Libraries
// EXEC IDMSRADM
```

*put SYSIDMS parameters, as required, here*

*/\**

*put IDMSRADM input parameters here*

*/\**

*/&*

Item	Description
IDMSLBLS	Name of the procedure provided at installation that contains the file definitions for CA IDMS dictionaries and databases. <b>Note:</b> IDMSLBLS, distributed on the installation media, should be tailored to define site-specific files. The proc may be copied and modified under another name so that one set of statements could be invoked for central version execution (include a SYSCtl definition and journal file definitions, for example) and another set of statements could be invoked for local mode execution. For a complete listing of IDMSLBLS, see IDMSLBLS Procedure later in this section.
radmarc	Filename of the sequential file
radm.archive	File-ID of the sequential file
nnnnnn	Volume serial id
ssss	Starting track or block of disk extent
tttt	Number of tracks
sys004	Logical unit assignment of the sequential file
sort.work1	File name of the sort work 1 file
sort.work2	File name of the sort work 2 file <b>Note:</b> If executing IDMSRADM for the purpose of reporting on space utilization, two sort work files must be defined.
sysnnn	Logical unit assignment for first sort work file
CA IDMS nn.n libraries	CA IDMS libraries established at installation

**Note:** SYSIDMS is a parameter file added to the JCL stream of CA IDMS batch jobs that allows you to specify runtime directives such as, the name of the DMCL and database to use at runtime, as well as operating system-dependent file information. SYSIDMS is defined in the IDMSLBLS procedure.

## z/VSE—Central Version (CV)

### JCL to execute IDMSRADM in CV mode (z/VSE)

```
// UPSI  b
// DLBL  sysctl,'cdms.sysctl',,DA
// EXTENT sysnnn,nnnnnn
// ASSGN sysnnn,DISK,VOL=nnnnnn,SHR
```

Item	Description
b	A 1 through 8 character UPSI bit switch, as specified in the IDMSOPTI module. <b>Note:</b> If IDMSRADM is linked with an IDMSOPTI module which includes an UPSI specification and a SYSCTL file is not defined in the proc or JCL, you must insert the UPSI statement at the beginning of the JCL (before the EXEC PROC=) and execute a tailored IDMSLBLS proc which includes DLBL and EXTENT statements for journal files.
sysctl	Filename of the SYSCTL file
cdms.sysctl	File-ID of the SYSCTL file. <b>Note:</b> A SYSCTL statement defined in the JCL (in-line or within the proc) is used to override the IDMSOPTI specification.
sysnnn	Logical unit assignment of the SYSCTL file
nnnnnn	Volume serial number

**Note:** For more information about SYSIDMS, see *CA IDMS Common Facilities Guide* parameters.

## z/VM—Local Mode

### JCL to execute IDMSRADM in local mode ('z/VM')

```
FILEDEF userdb DISK user db f (RECFM F LRECL pppp BLKSIZE pppp XTENT nnnn
Additional database, dictionary file assignments, as required
```

```
FILEDEF SYS004 TAP1 SL VOLID nnnnnn (BLKSIZE 5492
FILEDEF SYSLST PRINTER
FILEDEF sysjrn1 TAP1 SL VOLID nnnnnn (RECFM VB LRECL 111 BLKSIZE bbb
FILEDEF SYSIDMS DISK sysidms input a
FILEDEF SYSIPT DISK radm input a
GLOBAL LOADLIB dbalib
GLOBAL LOADLIB idmslib
OSRUN IDMSRADM
```

Item	Description
userdb	Ddname of the user database file
user db f	File identifier of the user database file
nnnnnn	Volume serial number of the sequential file which can be output to disk and is not required for a DELETE operation
5492	Block size of the sequential file which must be a multiple of 5492
sysjrn1	File identifier of local mode journal file
sysidms input a	File identifier of the file containing the input parameters for SYSIDMS
radm input a	File identifier of the file containing the IDMSRADM input parameters
dbalib	Filename of load library containing DMCL and DBname table
idmslib	Filename of CA IDMS/DB load library containing subschema and DMCL load modules

## z/VM—Central Version (CV)

To execute IDMSRADM in CV mode, modify the local mode JCL listed above as follows:

- Remove SYSJRNL and dictionary FILEDEF commands
- Identify the DC/UCF system to be accessed by IDMSRADM using the CVMACH and CVNUM SYSIDMS parameters

### Runtime Parameters

IDMSFD references the SYSIDMS file, a file in which you can specify parameters that describe physical requirements (such as DMCL or dictionary to access), runtime parameters, or operating system-specific file information.

For information on all SYSIDMS parameters, see *CA IDMS Common Facilities Guide* or *CA IDMS Navigational DML Programming Guide*.

### Creating the SYSIPT File

To create the SYSIPT file, enter these CMS commands:

```
XEDIT sysipt data a (NOPROF
INPUT
.
.
.
Enter IDMSRADM parameters, then press ENTER to end input
.
.
.
FILE
```

## IDMSLBS Procedure

### What is the IDMSLBS Procedure

IDMSLBS is a procedure provided during an CA IDMS z/VSE installation. It contains file definitions for the CA IDMS components listed below. These components are provided during installation:

- Dictionaries
- Sample databases
- Disk journal files
- SYSIDMS file

Tailor the IDMSLBS procedure to reflect the filenames and definitions in use at your site and include this procedure in z/VSE JCL job streams.

The sample z/VSE JCL provided in this document includes the IDMSLBS procedure. Therefore, individual file definitions for CA IDMS dictionaries, sample databases, disk journal files, and SYSIDMS file are not included in the sample JCL.

### IDMSLBS Procedure Listing

```
* ----- LIBDEFS -----
// LIBDEF *,SEARCH=idmslib.sublib
// LIBDEF *,CATALOG=user.sublib
/* ----- LABELS -----
// DLBL idmslib,'idms.library',2099/365
// EXTENT ,nnnnnn,,ssss,1500
// DLBL dccat,'idms.system.dccat',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,31
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dccatl,'idms.system.dccatlod',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,6
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dccatx,'idms.system.dccatx',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,11
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dcdml,'idms.system.dcdml',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,101
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dclod,'idms.system.dclod',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,21
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dclog,'idms.system.dclodlog',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,401
// ASSGN SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL dcrun,'idms.system.dclodrun',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,68
```

```
// ASSGN  SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL   dcscr,'idms.system.ddldcscr',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,135
// ASSGN  SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL   dmsg,'idms.sysmsg.ddldcmsg',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,201
// ASSGN  SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL   dclscr,'idms.sysloc.ddlocscr',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,6
// ASSGN  SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL   dirldb,'idms.sysdirl.ddldml',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,201
// ASSGN  SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL   dirllod,'idms.sysdirl.ddldclod',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,2
// ASSGN  SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL   empdemo,'idms.empdemo1',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,11
// ASSGN  SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL   insdemo,'idms.insdemo1',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,6
// ASSGN  SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL   orgdemo,'idms.orgdemo1',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,6
// ASSGN  SYSnnn,DISK,VOL=nnnnnn,SHR

// DLBL   empldem,'idms.sqldemo.empldemo',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,11
// ASSGN  SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL   infodem,'idms.sqldemo.infodemo',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,6
// ASSGN  SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL   projdem,'idms.projseg.projdemo',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,6
// ASSGN  SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL   indxdem,'idms.sqldemo.indxdemo',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,6
// ASSGN  SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL   sysctl,'idms.sysctl',2099/365,SD
// EXTENT SYSnnn,nnnnnn,,ssss,2
// ASSGN  SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL   secdd,'idms.sysuser.ddlsec',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,26
// ASSGN  SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL   dictdb,'idms.appldict.ddldml',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,51
// ASSGN  SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL   dloddb,'idms.appldict.ddldclod',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,51
// ASSGN  SYSnnn,DISK,VOL=nnnnnn,SHR
```



```

// DLBL    sqldd, 'idms.syssql.ddlcat', 2099/365, DA
// EXTENT  SYSnnn, nnnnnn, , , ssss, 101
// ASSGN   SYSnnn, DISK, VOL=nnnnnn, SHR
// DLBL    sqlld, 'idms.syssql.ddlcatl', 2099/365, DA
// EXTENT  SYSnnn, nnnnnn, , , ssss, 51
// ASSGN   SYSnnn, DISK, VOL=nnnnnn, SHR
// DLBL    sqlxdd, 'idms.syssql.ddlcatx', 2099/365, DA
// EXTENT  SYSnnn, nnnnnn, , , ssss, 26
// ASSGN   SYSnnn, DISK, VOL=nnnnnn, SHR
// DLBL    asfdml, 'idms.asfdict.ddldml', 2099/365, DA
// EXTENT  SYSnnn, nnnnnn, , , ssss, 201
// ASSGN   SYSnnn, DISK, VOL=nnnnnn, SHR
// DLBL    asflod, 'idms.asfdict.asflod', 2099/365, DA
// EXTENT  SYSnnn, nnnnnn, , , ssss, 401
// ASSGN   SYSnnn, DISK, VOL=nnnnnn, SHR
// DLBL    asfdata, 'idms.asfdict.asfdata', 2099/365, DA
// EXTENT  SYSnnn, nnnnnn, , , ssss, 201
// ASSGN   SYSnnn, DISK, VOL=nnnnnn, SHR
// DLBL    ASFDEFN, 'idms.asfdict.asfdefn', 2099/365, DA
// EXTENT  SYSnnn, nnnnnn, , , ssss, 101
// ASSGN   SYSnnn, DISK, VOL=nnnnnn, SHR
// DLBL    j1jrnL, 'idms.j1jrnL', 2099/365, DA
// EXTENT  SYSnnn, nnnnnn, , , ssss, 54
// ASSGN   SYSnnn, DISK, VOL=nnnnnn, SHR
// DLBL    j2jrnL, 'idms.j2jrnL', 2099/365, DA
// EXTENT  SYSnnn, nnnnnn, , , ssss, 54
// ASSGN   SYSnnn, DISK, VOL=nnnnnn, SHR

// DLBL    j3jrnL, 'idms.j3jrnL', 2099/365, DA
// EXTENT  SYSnnn, nnnnnn, , , ssss, 54
// ASSGN   SYSnnn, DISK, VOL=nnnnnn, SHR
// DLBL    SYSIDMS, '#SYSIPT', 0, SD
/+
/*

```

Item	Description
<i>idmslib.sublib</i>	Name of the sublibrary within the library containing CA IDMS modules
<i>user.sublib</i>	Name of the sublibrary within the library containing user modules
<i>idmslib</i>	Filename of the file containing CA IDMS modules
<i>idms.library</i>	File-ID associated with the file containing CA IDMS modules
<i>SYSnnn</i>	Logical unit of the volume for which the extent is effective
<i>nnnnnn</i>	Volume serial identifier of appropriate disk volume

<b>Item</b>	<b>Description</b>
ssss	Starting track (CKD) or block (FBA) of disk extent
dccat	Filename of the system dictionary catalog (DDLDCAT) area
idms.system.dccat	File-ID of the system dictionary catalog (DDLDCAT) area
dccatl	Filename of the system dictionary catalog load (DDLDCATL) area
idms.system.dccatlod	File-ID of the system dictionary catalog load (DDLDCATL) area
dccatx	Filename of the system dictionary catalog index (DDLDCATX) area
idms.system.dccatx	File-ID of the system dictionary catalog index (DDLDCATX) area
dcdml	Filename of the system dictionary definition (DDLDCML) area
idms.system.ddldml	File-ID of the system dictionary definition (DDLDCML) area
dclod	Filename of the system dictionary definition load (DDLDCLOD) area
idms.system.ddldclod	File-ID of the system dictionary definition load (DDLDCLOD) area
dclog	Filename of the system log area (DDLDCLOG) area
idms.system.ddldclog	File-ID of the system log (DDLDCLOG) area
dcrun	Filename of the system queue (DDLDCRUN) area
idms.system.ddldcrun	File-ID of the system queue (DDLDCRUN) area
dcscr	Filename of the system scratch (DDLDCSCR) area
idms.system.ddldcscr	File-ID of the system scratch (DDLDCSCR) area
dcmsg	Filename of the system message (DDLDCMSG) area
idms.sysmsg.ddldcmsg	File-ID of the system message (DDLDCMSG) area
dclscr	Filename of the local mode system scratch (DDLDCSCR) area
idms.sysloc.ddldcscr	File-ID of the local mode system scratch (DDLDCSCR) area
dirldb	Filename of the IDMSDIRL definition (DDLDCML) area
idms.sysdirl.ddldml	File-ID of the IDMSDIRL definition (DDLDCML) area
dirllod	Filename of the IDMSDIRL definition load (DDLDCLOD) area
idms.sysdirl.dirllod	File-ID of the IDMSDIRL definition load (DDLDCLOD) area
empdemo	Filename of the EMPDEMO area
idms.empdemo1	File-ID of the EMPDEMO area

<b>Item</b>	<b>Description</b>
insdemo	Filename of the INSDEMO area
idms.insdemo1	File-ID of the INSDEMO area
orgdemo	Filename of the ORGDemo area
idms.orgdemo1	File-ID of the ORDDemo area
empldem	Filename of the EMPLDemo area
idms.sqldemo.empldemo	File-ID of the EMPLDemo area
infodem	Filename of the INFODemo area
idms.sqldemo.infodem	File-ID of the INFODemo area
projdem	Filename of the PROJDemo area
idms.projseg.projdemo	File-ID of the PROJDemo area
indxdem	Filename of the INDXDemo area
idms.sqldemo.indxdem	File-ID of the INDXDemo area
sysctl	Filename of the SYSCTL file
idms.sysctl	File-ID of the SYSCTL file
secdd	Filename of the system user catalog (DDLSEC) area
idms.sysuser.ddlsec	File-ID of the system user catalog (DDLSEC) area
dictdb	Filename of the application dictionary definition area
idms.appldict.ddldml	File-ID of the application dictionary definition (DDLDDL) area
dloddb	Filename of the application dictionary definition load area
idms.appldict.ddldclod	File-ID of the application dictionary definition load (DDLDCLOD) area
sqldd	Filename of the SQL catalog (DDLDCAT) area
idms.syssql.ddlcat	File-ID of the SQL catalog (DDLDCAT) area
sqllod	Filename of the SQL catalog load (DDLDCATL) area
idms.syssql.ddlcatl	File-ID of SQL catalog load (DDLDCATL) area
sqlxdd	Filename of the SQL catalog index (DDLDCATX) area
idms.syssql.ddlcatx	File-ID of the SQL catalog index (DDLDCATX) area
asfdml	Filename of the asf dictionary definition (DDLDDL) area

<b>Item</b>	<b>Description</b>
idms.asfdict.ddldml	File-ID of the asf dictionary definition (DDLML) area
asflod	Filename of the asf dictionary definition load (ASFLOD) area
idms.asfdict.asflod	File-ID of the asf dictionary definition load (ASFLOD) area
asfdata	Filename of the asf data (ASFDATA) area
idms.asfdict.asfdata	File-ID of the asf data area (ASFDATA) area
ASFDEFN	Filename of the asf data definition (ASFDEFN) area
idms.asfdict.asfdefn	File-ID of the asf data definition area (ASFDEFN) area
j1jrnl	Filename of the first disk journal file
idms.j1jrnl	File-ID of the first disk journal file
j2jrnl	Filename of the second disk journal file
idms.j2jrnl	File-ID of the second disk journal file
j3jrnl	Filename of the third disk journal file
idms.j3jrnl	File-ID of the third disk journal file
SYSIDMS	Filename of the SYSIDMS parameter file

# Chapter 19: MAINTAIN ASF

---

This section contains the following topics:

- [Description](#) (see page 301)
- [Usage](#) (see page 305)
- [JCL](#) (see page 310)
- [Examples](#) (see page 317)

## Description

### Purpose

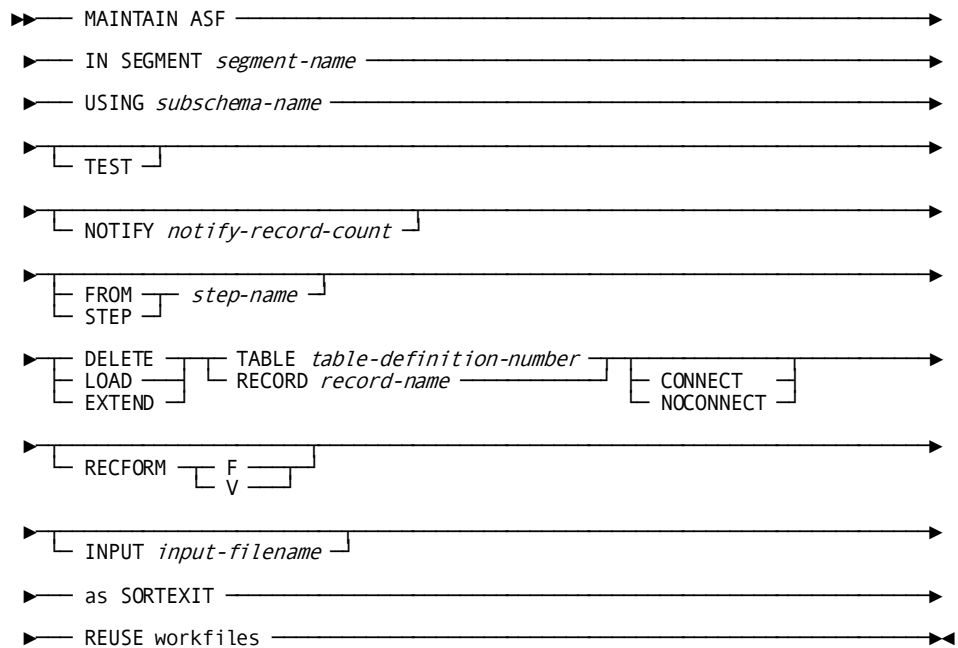
The MAINTAIN ASF utility statement deletes, loads, or extends an ASF-defined table.

You can run MAINTAIN ASF all at once or break it into steps.

### Authorization

You must have DBAWRITE authority on the area(s) containing the table and its associated index(es).

### Stick.Syntax



**Parameters**

**IN SEGMENT**

Specifies the segment containing the table to be processed.

***segment-name***

The name of the segment.

**USING**

Specifies the subschema that defines the table.

***subschema-name***

The name of the subschema.

**TEST**

Processes only the first 100 records of the database.

See "Usage", below.

**NOTIFY**

Directs MAINTAIN ASF to send a message to the operator whenever a specified number of rows have been processed.

The message states the step currently being executed and the number of rows that have been processed.

***notify-record-count***

The number of records to process before sending a message.

**FROM**

Specifies that MAINTAIN ASF should begin at a specified step and complete all remaining steps.

***step-name***

The name of the first step that MAINTAIN ASF should execute.

If the specified restart step is any step except IDMSTABX, all other parameters are ignored. In this case, MAINTAIN ASF gets the information normally provided by the other parameters from the intermediate work files.

**STEP**

Directs MAINTAIN ASF to execute only one step of the table maintenance process.

By default, if you do not specify STEP, MAINTAIN ASF will perform all steps.

If the specified step is any step except IDMSTABX, all other parameters are ignored. In this case, MAINTAIN ASF gets the information normally provided by the other parameters from the intermediate work files.

***step-name***

The name of the first or only step to execute.

The name must be one of the following:

- IDMSTABX
- SORT3
- IDMSDBL3
- SORT4
- IDMSDBL4

#### **DELETE**

Directs MAINTAIN ASF to erase all rows of the specified table, and all associated index structures. The table definition remains intact.

#### **LOAD**

Directs MAINTAIN ASF to load the specified table from an external file.

All indexes associated with the table must be empty.

#### **EXTEND**

Directs MAINTAIN ASF to add rows to the existing specified table.

At least one index must not be empty. If all indexes are empty, MAINTAIN ASF will perform a LOAD.

#### **TABLE**

Specifies the table with its table definition number.

***"table-definition-number"***

The table definition number assigned to the table when it was created.

The table definition must be enclosed in double quotes.

#### **RECORD**

Specifies the table with its record name.

***"record-name"***

The record name by which the table is identified in the schema.

The record name must be enclosed in double quotes.

### **CONNECT**

Directs MAINTAIN ASF to connect all rows added during a LOAD or EXTEND operation to all manual and automatic indexes.

By default, if you do not specify CONNECT or NOCONNECT, MAINTAIN ASF will connect added rows to the indexes.

### **NOCONNECT**

Directs MAINTAIN ASF not to build index structures during the LOAD or EXTEND operation.

If you specify NOCONNECT, you must use the MAINTAIN INDEX utility at a later time to build the index(es).

For more information on the MAINTAIN INDEX utility, see *CA IDMS Utilities Guide*

### **RECFORM**

Specifies whether the input file has a fixed or variable format.

By default, if you do not specify a format, MAINTAIN ASF assumes the file has a fixed format.

#### **F**

Specifies that the file has a fixed format.

This is the default.

#### **V**

Specifies that the file has a variable format.

### **INPUT *input-filename***

Specifies the DD/ASSIGN/FILEDEF/FILE statement in the JCL that contains the input for a LOAD or EXTEND operation.

By default, if you do not specify a DD/ASSIGN/FILEDEF/FILE, MAINTAIN ASF uses SYS001.

### **as SORTEXIT**

Causes each DBLx step in the utility to return its input data directly from the preceding sort instead of having the sort write the data to a workfile. This option eliminates one workfile for each sort and saves the I/O it takes to write then read the workfile.

### **REUSE workfiles**

Causes each step in the utility to reuse an existing workfile, if possible, when writing its output data, instead of writing to a new one for each step. This reduces the number of workfiles that need to be allocated.



---

## Usage

### How to Submit the MAINTAIN ASF Statement

You submit the MAINTAIN ASF statement to CA IDMS/DB only through the batch command facility. You must be running CA IDMS/DB in local mode.

All affected areas must be varied offline to all central versions.

### Run ASFOOAKD Dialog after Load or Delete

After loading a table or deleting a table that will be reloaded by some other means, you must run the ASFOOAKD dialog.

Use the task code ASFOOAKD to invoke the dialog, and supply the table definition number in response to the TABLE DEFINITION NUMBER prompt.

ASFOOAKD performs the following functions:

- If an RFUR-*nnnnnn*-OOAK record exists for the table, ASFOOAKD updates the time/date stamp in the record and saves the database key of the record as data in the RDEFREC occurrence associated with the table.
- If no RFUR-*nnnnnn*-OOAK record exists for the table, ASFOOAKD stores an RFUR-*nnnnnn*-OOAK record and saves the database key of the record as data in the RDEFREC occurrence associated with the table.

### When to Specify TEST

Specify TEST if you want to verify the results of the operation without running through it completely.

**Important!** Specify TEST only when loading or extending a table into a test database. The incomplete processing performed is likely to corrupt a working database.

### SORTEXIT and FROM/STEP

When using the FROM and STEP options with the SORTEXIT option, each pair of SORT*n* and DBL*x* steps are considered to be one step. If either half of the SORT*n*/DBL*x* is specified on a FROM or STEP option, processing will start with the SORT*n* step and the DBL*x* step will also be executed. For example:

- FROM IDMSDBL3 will start with step SORT3 and will continue to the end.
- STEP SORT3 will run steps SORT3 and IDMSDBL3.

### SORTEXIT/REUSE WORKFILE Restart Considerations

Since SORTEXIT combines each SORT $n$  step with the DBL $x$  step that follows it, if a failure occurs in the DBL $x$  step, a restart (if a restart is possible) must begin with the sort step and the input to the step will be resorted. Non-SORTEXIT mode will take longer to run but can be restarted after the sort in this case. Therefore, if restart time is more critical than normal runtime do not run the utility as a sortexit.

If the REUSE WORKFILE option is used with SORTEXIT, some input workfiles will be used as output files in the same step. Therefore if these two options are used together and a failure occurs, the utility must be restarted from the beginning.

## The Steps of MAINTAIN ASF

MAINTAIN ASF consists of the following steps which you can run separately or as a single operation:

Step	Description
IDMSTABX	<ul style="list-style-type: none"><li>■ Interprets the control statements</li><li>■ Reads the database</li><li>■ Writes index descriptors to the SYS003 file</li><li>■ Writes sort control parameters to the SYSPCH file</li></ul>
SORT3	Sorts the contents of the SYS003 file and puts the results in SYS004.
IDMSDBL3	<ul style="list-style-type: none"><li>■ Erases old index structures, if there are any</li><li>■ Builds the new index structures</li><li>■ Establishes pointers for user-owned index sets, and puts them in the SYS005 file</li></ul>
SORT4	Sorts the contents of the SYS005 file, and puts the results in SYS006.
IDMSDBL4	Fills in prefix pointers for user-owned indexes

**Each Step has Input and Output**

This table describes the input and output files if MAINTAIN ASF is executed without the SORTEXIT and REUSE options. The effect of those options is shown under "JCL" later in this chapter.

Step	Input	Output
IDMSTABX	<ul style="list-style-type: none"> <li>■ Control parameters from your submission of the utility</li> <li>■ The database</li> </ul>	<ul style="list-style-type: none"> <li>■ SYS003 contains index membership information</li> <li>■ SYSPCH contains sort parameters</li> </ul>
SORT3	<ul style="list-style-type: none"> <li>■ SYS003 from IDMSTABX</li> <li>■ SYS002 contains the sort parameters from IDMSTABX</li> </ul>	SYS004 contains the sorted contents of SYS003
IDMSDBL3	SYS004 from SORT3	<ul style="list-style-type: none"> <li>■ SYS005 contains prefix pointer information for user-owned indexes</li> <li>■ SYSPCH contains sort parameters</li> </ul>
SORT4	<ul style="list-style-type: none"> <li>■ SYS005 from IDMSDBL3</li> <li>■ SYS002 contains the sort parameters from IDMSDBL3</li> </ul>	SYS006 contains the sorted contents of SYS005
IDMSDBL4	SYS006 from SORT4	SYSLST contains a summary of the results of the MAINTAIN ASF operation

### Sort Output after Each Step

If you run MAINTAIN ASF a step at a time, you must use the sort parameters to sort the contents of the intermediate work files. You can use your own sort program or IDMSORT.

Sort the intermediate work files as follows:

Sort name	File to sort	Sort order	Sort on	Begins at
SORT3	SYS003	Ascending	36 bytes	Byte 5
SORT4	SYS005	Ascending	28 bytes	Byte 5

**z/VSE users:** The sort parameters generated by IDMSTABX and IDMSDBL3 are not sufficient for stand alone sort programs running under z/VSE. If you want to use your own sort program, you must add "WORK=" parameters to specify more than one sort work file.

## Sample Output

Below is a sample report run with MAINTAIN ASF:

```
IDMSBCF                                IDMS Batch Command Facility                mm/dd/yy  PAGE  1
SET BATCH WIDTH PAGE 80;
Status = 0
MAINTAIN ASF IN SEGMENT ASFDICT USING RJ000114
  EXTEND TABLE "114"          INPUT ASFIPT01 ;
UT010002 BEGINNING PROCESSING FOR STEP IDMSTABX
UT012001 IDMSTABX RELEASE nn.n TAPE volser STARTED
UT012003 8 RECORDS WERE STORED IN THE DATABASE
UT012004 8 INTERMEDIATE RECORDS WERE WRITTEN TO SYS003
UT012006 SYS003  MAXIMUM RECORD SIZE IS 40
UT012007 IDMSTABX RELEASE nn.n PROCESSING COMPLETED
UT010003 STEP IDMSTABX HAS COMPLETED SUCCESSFULLY
UT010002 BEGINNING PROCESSING FOR STEP SORT3
UT009001 IDMSDBLY RELEASE nn.n TAPE volser SORT STARTED
UT009002 8 RECORDS WERE READ FROM SYS003
UT009003 8 RECORDS WERE WRITTEN TO SYS004
UT009004 IDMSDBLY RELEASE nn.n SORT COMPLETED SUCCESSFULLY
UT010003 STEP SORT3  HAS COMPLETED SUCCESSFULLY
UT010002 BEGINNING PROCESSING FOR STEP IDMSDBL3
UT006001 IDMSDBL3 RELEASE nn.n TAPE volser PROCESSING STARTED
UT006007 8 INTERMEDIATE RECORDS WERE READ FROM SYS004
UT006002 7 INTERMEDIATE RECORDS WERE WRITTEN TO SYS005
UT006005 NO DATABASE ERRORS WERE ENCOUNTERED
UT006006 IDMSDBL3 RELEASE nn.n PROCESSING COMPLETED
UT010003 STEP IDMSDBL3 HAS COMPLETED SUCCESSFULLY
UT010002 BEGINNING PROCESSING FOR STEP SORT4
UT009001 IDMSDBLY RELEASE nn.n TAPE volser SORT STARTED
UT009002 7 RECORDS WERE READ FROM SYS005
UT009003 7 RECORDS WERE WRITTEN TO SYS006
UT009004 IDMSDBLY RELEASE nn.n SORT COMPLETED SUCCESSFULLY
UT010003 STEP SORT4  HAS COMPLETED SUCCESSFULLY
UT010002 BEGINNING PROCESSING FOR STEP IDMSDBL4
UT007001 IDMSDBL4 RELEASE nn.n TAPE volser PROCESSING STARTED
UT007004 NO DATABASE ERRORS WERE ENCOUNTERED
UT007005 NO LOGIC ERRORS WERE ENCOUNTERED
UT007002 7 RECORDS WERE READ FROM SYS006
UT007006 IDMSDBL4 RELEASE nn.n PROCESSING COMPLETED
UT010003 STEP IDMSDBL4 HAS COMPLETED SUCCESSFULLY
UT013001 DATABASE INDEX/ASF TABLE HAS COMPLETED SUCCESSFULLY
Status = 0

AutoCommit will COMMIT transaction

Command Facility ended with no errors or warnings
```

## JCL

In the sections to follow is JCL for CA IDMS/DB running in local mode for these operating systems:

- z/OS
- z/VSE
- z/VM

### JCL Considerations

When you submit a MAINTAIN ASF statement to CA IDMS/DB through the batch command facility, the JCL to execute the facility must include statements to define:

- The files containing the areas to be processed
- Sort space
- The intermediate work files

The following table indicates which workfiles are created and read by the different utility steps depending on the use of the SORTEXIT and REUSE WORKFILE options.

Step	Input	Output
MAINTAIN ASF: NOT Sortexit mode and NOT reusing workfiles.		
IDMSTABX		SYS003
SORT3	SYS003	SYS004
IDMSDBL3	SYS004	SYS005
SORT4	SYS005	SYS006
IDMSDBL4	SYS006	
MAINTAIN ASF: NOT Sortexit mode and REUSING workfiles.		
IDMSTABX		SYS003
SORT3	SYS003	SYS004
IDMSDBL3	SYS004	SYS003
SORT4	SYS003	SYS004
IDMSDBL4	SYS004	
MAINTAIN ASF: SORTEXTIT mode and NOT reusing workfiles.		

Step	Input	Output
IDMSTABX		SYS003
SORT3/IDMSDBL3	SYS003	SYS005
SORT4/IDMSDBL4	SYS005	
MAINTAIN ASF: SORTEXTIT mode and REUSING workfiles.		
IDMSTABX		SYS003
SORT3/IDMSDBL3	SYS003	SYS003
SORT4/IDMSDBL4	SYS003	

**Note:** For sample JCL to run the Batch Command Facility, see *CA IDMS Utilities Guide*.

## z/OS

### DD statements for the batch command facility (z/OS)

```
//asflod DD DSN=idms.asfdict.asflod,DISP=SHR ← -- Only needed if the 'RU'
//userdb DD DSN=userdb,DISP=SHR          subschema is in the ASF load
exec.
//RELDCTL DD DSN=reldctl.file,DISP=(NEW,PASS),UNIT=disk,
//          DCB=(RECFM=FB,LRECL=60,BLKSIZE=bbb)
//infile  DD DSN=input.file,DISP=OLD
//SYS003  DD DSN=user.db1003,DISP=(NEW,PASS),UNIT=tape003,
//          DCB=(RECFM=VB,LRECL=rrr003,BLKSIZE=bbb003)
//SYS004  DD DSN=user.db1004,DISP=(NEW,PASS),UNIT=tape004,
//          DCB=(RECFM=VB,LRECL=rrr004,BLKSIZE=9bb004)
//SYS005  DD DSN=user.db1005,DISP=(NEW,PASS),UNIT=tape005,
//          DCB=(RECFM=VB,LRECL=rrr005,BLKSIZE=bbb005)
//SYS006  DD DSN=user.db1006,DISP=(NEW,PASS),UNIT=tape006,
//          DCB=(RECFM=VB,LRECL=rrr006,BLKSIZE=bbb006)
//SYSPCH  DD DSN=&&sort,DISP=(NEW,PASS),UNIT=disk,
//          SPACE=(TRK,1),DCB=BLKSIZE=80
```

Item	Description
userdb	Ddname of the existing database file
user.userdb	Data set name of the existing database file
reldctl.file	File containing control records; the logical record length of this file must be a multiple of four.

<b>Item</b>	<b>Description</b>
bbb	Block size of the RELDCTL file
disk	Symbolic device name of the RELDCTL file
infile	Ddname of the file containing input data for the table; If the records are a fixed length, the length must match the record length as defined in the subschema.
input.file	Data set name of the file containing input data for the table
user.dbl003	Data set name of the intermediate work file containing index descriptors from IDMSTABX
tape003	Symbolic device name of the SYS003 file
rrr003	Record size of the SYS003 file
bbb003	Block size of the SYS003 file
user.dbl004	Data set name of the intermediate work file containing the output from SORT3
tape004	Symbolic device name of the SYS004 file
rrr004	Record size of the SYS004 file; should be the same as rrr003
bbb004	Block size of the SYS004 file; should be the same as bbb003
user.dbl005	Data set name of the intermediate work file containing pointers for user owned index sets from IDMSDBL3
tape005	Symbolic device name of the SYS005 file
rrr005	Record size of the SYS005 file
bbb005	Block size of the SYS005 file
user.dbl006	Data set name of the intermediate work file containing sorted pointers for user owned index sets from SORT4
tape006	Symbolic device name of the SYS006 file
rrr006	Record size of the SYS006 file; should be the same as rrr005
bbb006	Block size of the SYS006 file; should be the same as bbb005
&.&sort.	Data set name of the SYSPCH file containing sort parameters from IDMSTABX and IDMSDBL3
disk	Symbolic device name of the SYSPCH file



## z/VSE—Local Mode

**Note:** MAINTAIN ASF may need to be relink edited with an IDMSDBLP module before being executed.

For information on the IDMSDBLP module, see the 'z/VSE JCL' chapter in *CA IDMS Utilities Guide*.

### JCL to execute MAINTAIN ASF in local mode (z/VSE)

```
// EXEC PROC=IDMSLBS
// DLBL RELDCTL,'user.reldctl',,SD
// EXTENT SYSnnn,nnnnn,,ssss,tttt
// ASSGN SYSnnn,DISK,VOL=vvvvvv,SHR
// DLBL infile,'input.data',,SD
// EXTENT SYS001,nnnnn,,ssss,tttt
// ASSGN SYS001,DISK,VOL=vvvvvv,SHR
// DLBL SYSPCH,'sort.info'
// EXTENT SYSPCH,nnnnn,,ssss,tttt
// ASSGN SYSPCH,DISK,VOL=nnnnnn,SHR
// DLBL SYS003,'user.fil3',,SD
// EXTENT SYS003,nnnnn,,ssss,tttt
// ASSGN SYS003,DISK,VOL=nnnnnn,SHR
// DLBL SYS004,'user.fil4',,SD
// EXTENT SYS004,nnnnn,,ssss,tttt
// ASSGN SYS004,DISK,VOL=nnnnnn,SHR
// DLBL SYS005,'user.fil5',,SD
// EXTENT SYS005,nnnnn,,ssss,tttt
// ASSGN SYS005,DISK,VOL=nnnnnn,SHR
// DLBL SYS006,'user.fil6',,SD
// EXTENT SYS006,nnnnn,,ssss,tttt
// ASSGN SYS006,DISK,VOL=nnnnnn,SHR
// DLBL SORTWK1,'sort.work.file'
// EXTENT SYSnnn,SYSWK0,,1,30
// ASSGN SYSnnn,SYS001,OUTPUT
```

*Put additional sort files, as required, here*

```
// DLBL SYSIDMS,'#SYSIPT'
// EXEC IDMSBCF
Put parms for SYSIDMS here
/*
Put parms for MAINTAIN ASF here
/*
/&
```

Item	Description
IDMSLBLS	Name of the procedure provided at installation that contains the file definitions for CA IDMS dictionaries and databases. <b>Note:</b> Tailor the IDMSLBLS proc provided on the installation media to include the appropriate files (database, journal, etc.) for local mode processing.
user.reldctl	File containing control records <b>Note:</b> The logical record length of this file must be a multiple of four.
SYSnnn	Logical unit of the volume for which the extent is effective
nnnnnn	Volume serial ID
ssss	Starting track of block of disk extent
tttt	Number of tracks
infile	Filename of the input data for LOAD or EXTEND operations as specified in MAINTAIN ASF input statements. If not named specifically, filename should be SYS001.
input.data	File-ID of the input data for LOAD or EXTEND operations
sort.info	sort control info for intermediate sorts which is punched by the utility
user.fil3	Intermediate file written by utility steps
user.fil4	Intermediate file written by utility steps
user.fil5	Intermediate file written by utility steps
user.fil6	Intermediate file written by utility steps

**Note:** SYSIDMS is a parameter file added to the JCL stream of CA IDMS batch jobs that allows you to specify runtime directives such as, the name of the DMCL and database to use at runtime, as well as operating system-dependent file information. SYSIDMS is defined in the IDMSLBLS procedure.

For more information about SYSIDMS, see *CA IDMS Common Facilities Guide* parameters.

## z/VM—Local Mode

### JCL to execute MAINTAIN ASF in local mode ('z/VM')

```
FILEDEF RELDCTL DISK user reldctl a (RECFM FB LRECL 60 BLKSIZE bbbctl)
FILEDEF SYS001 DISK input data a (RECFM FB LRECL rrr BLKSIZE bbb)
FILEDEF SYSPCH DISK sort maint a (RECFM F BLKSIZE 80)
FILEDEF SYS003 DISK tabx out a (RECFM VB LRECL rrr BLKSIZE bbb)
FILEDEF SYS004 DISK sort1 out a (RECFM VB LRECL rrr BLKSIZE bbb)
FILEDEF SYS005 DISK dbl3 out a (RECFM VB LRECL rrr BLKSIZE bbb)
FILEDEF SYS006 DISK sort2 out a (RECFM VB LRECL rrr BLKSIZE bbb)
FILEDEF SYSIDMS DISK sysidms input a (RECFM FB LRECL rrr BLKSIZE bbb)
OSRUN IDMSBCF
```

Item	Description
<i>user reldctl a</i>	File containing control records. <b>Note:</b> The logical record length of this file must be a multiple of four.
<i>bbbctl</i>	Size of the RELDCTL file which should be a multiple of 60 with max size of 32,760
<i>input data a</i>	MAINTAIN ASF input statements
<i>rrr</i>	Size of input record which must agree with the record length as defined in subschema (SR51)
<i>bbb</i>	Blocksize
<i>sort maint a</i>	Sort control information punched by utility
<i>tabx out a</i>	Filename of the utility's intermediate output file
<i>sort1 out a</i>	Filename of the utility's intermediate output file
<i>dbl3 out a</i>	Filename of the utility's intermediate output file
<i>sort2 out a</i>	Filename of the utility's intermediate output file
<i>sysidms input a</i>	File containing SYSIDMS parameters
IDMSBCF	Program to be executed from the CMS LOADLIB

## z/VM—Central version (CV)

The CMS commands you use to run MAINTAIN ASF under the central version follow:

### JCL to execute MAINTAIN ASF in CV mode ('z/VM')

```
FILEDEF SYSIPT DISK sysipt data a (RECFM F LRECL ppp BLKSIZE ppp
EXEC IDMSFD
OSRUN IDMSBCF
```

Item	Description
sysipt data a	Filename, type, and mode of the file containing MAINTAIN ASF parameters
ppp	Record length of the <i>sysipt</i> data file
nnn	Block size of the <i>sysipt</i> data file
IDMSFD	Exec which defines all FILEDEFS, TXTLIBs, and LOADLIBs required by the system
IDMSBCF	Program to be executed from the CMS LOADLIB

### Runtime Parameters

IDMSFD references the SYSIDMS file, a file in which you can specify parameters that describe physical requirements (such as DMCL or dictionary to access), runtime parameters, or operating system-specific file information.

For more information about SYSIDMS, See *CA IDMS Common Facilities Guide* parameters.

### Creating the SYSIPT File

To create the SYSIPT file, enter these CMS commands:

```
XEDIT sysipt data a (NOPROF
INPUT
.
.
.
Enter MAINTAIN ASF parameters, then press ENTER to end input
.
.
.
FILE
```

---

## Examples

Examples of the MAINTAIN ASF input parameters follow:

### Example 1

The input parameters shown below direct MAINTAIN ASF to load the data in the input file as rows of the data table with table definition number 119. By default, MAINTAIN ASF connects only automatic indexes associated with the data table.

```
SUBSCHEMA=RU000119  
LOAD TDN=119
```

### Example 2

The input parameters shown below direct MAINTAIN ASF to load the data in the input file as rows of the data table with table definition number 183. Because CONNECT is specified, MAINTAIN ASF connects all indexes associated with the data table.

```
SUBSCHEMA=RU000183  
EXTEND TDN=183,CONNECT
```

### Example 3

The following command directs MAINTAIN ASF to load the table "114" using input read from ASFIPT01. The steps used to build the index on the table will run as a sortexit and will reuse workfiles:

```
MAINTAIN ASF IN SEGMENT ASFDICT USING RU000114  
AS SORTEXIT REUSE WORKFILES  
LOAD TABLE "114" INPUT ASFIPT01;
```



# Appendix A: Logical Record Facility

---

This section contains the following topics:

[Introduction](#) (see page 319)

[Checking Path Statuses](#) (see page 321)

[Examples](#) (see page 322)

## Introduction

The **Logical Record Facility (LRF)** is a runtime facility that allows application programmers to access CA IDMS data without having to know the physical structure of the database. Under LRF, programmers do not have to use database navigation statements to access information. This is because the DBA predefines database access logic that is typically coded by programmers. The following steps take place when you generate a table through ASF with LRF:

1. **Record and element definitions are added to the data dictionary** (stored tables only):
  - Records are named RFUR-*nnnnnn*-DATA, where *nnnnnn* is the table definition number.
  - Element names are the STORED COLUMN NAME shown on the Extended Column Definition screen.
2. **Record and set descriptions are added to the IDMS/R schema** (stored tables only). The following descriptions are added (*nnnnnn* is the table definition number):
  - **RFUR-*nnnnnn*-DATA record**—The record type in which rows of the table are stored as record occurrences.
  - **RFUR-*nnnnnn*-OOAK record**—An IDMS/R-defined OOAK record.
  - **RFUS-*nnnnnn*-SET set**—An unsorted indexed set defined if the storage sequence of the table is NEXT, PRIOR, FIRST, or LAST.
  - **RFPS-*nnnnnn*-SET set**—An indexed set defined if the storage sequence of the table is designated by the key number of an index key.
  - **RFFS-*nnnnnn*-yyyy set**—One for each fast access key defined on the Extended Key Definition screen. The variable *yyyy* is the field definition number for the column, which is assigned in the FDEFREC.
  - **RFKS-*nnnnnn*-zzzz set**—One for each additional key defined on the Extended Key Definition screen that is not used as the storage sequence. The variable *zzzz* is the key number of the additional key.

3. **A subschema is generated**, which consists of:
  - A **logical record** that contains the following elements:
    - **ASF-RDEF-REC**— The table definition record
    - **RFUR-nnnnnn-OOAK**—The system-defined OOAK record
    - **RFUR-nnnnnn-DATA**—The record used to store rows of the table
    - **RFUR-nnnnnn-WORK**—An IDD work record, which is included if any work fields are defined for the table

ASF names the logical record by modifying the table name to conform to COBOL naming conventions.
  - **Logical-record paths**, which allow an application program to access occurrences of the logical record:
    - For a stored table, ASF generates an OBTAIN, STORE, MODIFY, and ERASE path by default.
    - For a table definition (an ASF view), ASF generates an OBTAIN and MODIFY path only.
4. **A map is created**. The columns defined for the table become the data fields for the map. The map is named *RMnnnnnn* (*nnnnnn* is the table definition number).
5. **A dialog that uses the subschema and map is generated** to allow online access of the table through ASF. The dialog is named *RDnnnnnn* (*nnnnnn* is the table definition number). In addition to the subschema (*RUnnnnnn*) and map (*RMnnnnnn*), the dialog has the following components:
  - **ASF-PARM-REC**— A parameter record included in every ASF-generated dialog
  - **Process modules**, as follows:
    - **RDnnnnnn-PREMAP**—Premap process, which prepares the map for display
    - **RDnnnnnn-CLEAR**—Response process for the CLEAR key, which exits from the ASF screen
    - **RDnnnnnn-ENTER**—Response process for the ENTER key, which is used to retrieve the next table row
    - **RDnnnnnn-PA1**—Response process for the PA1 key, which exits from ASF
    - **RDnnnnnn-PF1**—Response process for [PF1], which stores a table row
    - **RDnnnnnn-PF2**—Response process for [PF2], which modifies a table row
    - **RDnnnnnn-PF3**—Response process for [PF3], which erases a table row

**Note:** For more information on table generation through ASF, see the chapter "Creating and Using Stored Tables".



**You can extend the capabilities of ASF by tailoring the subschemas it generates.** You do this by using the subschema compiler to change the ASF-generated paths. The following considerations apply:

- If you want to continue to access the subschema through ASF, you cannot change the way the RFUR-*nnnnn*-DATA records are accessed. You should leave intact the logic that uses the ASF-RDEF-OWN, ASF-RDEF-REC, RFUR-*nnnnn*-OOAK, and RFUR-*nnnnn*-DATA records.
- If you ever regenerate the table in ASF, your changes will not be reflected in the new table.
- If you use the subschema compiler to add fields (such as work fields) to an ASF-generated table, the fields won't be recognized by the IDB catalog. Therefore, it is best to add the fields through ASF and then modify them, as appropriate.
- If tables generated through ASF are placed in extent areas, the application that uses the subschema must ready the areas in a shared usage mode. This is done through the READY ALL statement.

**Note:** Be aware that when you change an ASF-generated subschema, you may also have to change the associated map and dialogs.

In the examples shown below, modifications are enclosed in a box.

## Checking Path Statuses

When accessing ASF tables you should always check for the following path statuses:

- **INVALID\_DATA** is returned when the data violates the definition-time selection criteria (for example, WHERE STATE EQ 'MA' and the program tries to replace the state with 'NY'). When **INVALID\_DATA** is returned, the **ERROR\_STATUS** field in the IDMS communications block is set to 0000.
- **DEFN\_MISSING** is returned when the record definition cannot be found. When **DEFN\_MISSING** is returned, the **ERROR\_STATUS** field in the IDMS communications block is set to 0000.
- **OOAK\_MISSING** is returned when a one-of-a-kind record cannot be found. When **OOAK\_MISSING** is returned, the **ERROR\_STATUS** field in the IDMS communications block is set to 0000.
- **SYNC\_ERROR** is returned when the time stamp in the catalog and the table definition do not match. When **SYNC\_ERROR** is returned the **ERROR-STATUS** field in the IDMS communications block is set to 0000.

The return of any of these statuses indicates a fatal error; for more information, consult your DBA.

## Examples

This section contains the following examples:

- Allowing for CALC database entry
- Adding path-groups to expand functionality
- Implementing referential integrity checks for a table
- Performing a difference join
- Calculating summary information

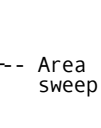
### Example 1—Allowing for CALC Database Entry

This example shows how to modify an ASF-generated table definition to allow for CALC database entry. The table definition joins the EMPLOYEE, EMPOSITION, and JOB records from the sample employee database. By default, the ASF-generated application performs an area sweep to retrieve the first record in the path.

**ASF-generated Subschema**

```
ADD
SUBSCHEMA NAME IS RJ000199 OF SCHEMA NAME IS IDMSR VERSION IS 1
.
.
ADD
LOGICAL RECORD NAME IS EMP-EMPOSITION
ELEMENTS ARE
  ASF-RDEF-REC
  EMPLOYEE
  EMPOSITION
  JOB
.
ADD
PATH-GROUP NAME IS MODIFY EMP-EMPOSITION
SELECT
  FIND CURRENT EMPLOYEE
    ON 0000 NEXT
  MODIFY EMPLOYEE
    ON 0000 NEXT
  FIND CURRENT EMPOSITION
    ON 0000 NEXT
  MODIFY EMPOSITION
    ON 0000 NEXT
  FIND CURRENT JOB
    ON 0000 NEXT
  MODIFY JOB
    ON 0000 NEXT
.
ADD
PATH-GROUP NAME IS OBTAIN EMP-EMPOSITION
SELECT
  FIND FIRST ASF-RDEF-OWN WITHIN IDMSR-AREA
    ON 0000 NEXT
    ON 0307 ITERATE
  OBTAIN EACH EMPLOYEE WITHIN EMP-DEMO-REGION
    ON 0000 NEXT
    ON 0307 ITERATE
  OBTAIN EACH EMPOSITION WITHIN EMP-EMPOSITION
    ON 0000 NEXT
    ON 0307 ITERATE
  OBTAIN OWNER WITHIN JOB-EMPOSITION
    ON 0000 NEXT
.

```



The diagram shows a bracket on the right side of the 'OBTAIN' statements in the 'OBTAIN EMP-EMPOSITION' path group, labeled 'Area sweep'. The bracket spans the following lines:

```
  OBTAIN EACH EMPLOYEE WITHIN EMP-DEMO-REGION
    ON 0000 NEXT
    ON 0307 ITERATE
  OBTAIN EACH EMPOSITION WITHIN EMP-EMPOSITION
    ON 0000 NEXT
    ON 0307 ITERATE
```

### Modified Subschema

```
ADD
SUBSCHEMA NAME IS RU000199 OF SCHEMA NAME IS IDMSR VERSION IS 1
.
.
ADD
LOGICAL RECORD NAME IS EMP-EMPOSITION
ELEMENTS ARE
  ASF-RDEF-REC
  EMPLOYEE
  EMPOSITION
  JOB
.
ADD
PATH-GROUP NAME IS MODIFY EMP-EMPOSITION
SELECT
  FIND CURRENT EMPLOYEE
    ON 0000 NEXT
  MODIFY EMPLOYEE
    ON 0000 NEXT
  FIND CURRENT EMPOSITION
    ON 0000 NEXT
  MODIFY EMPOSITION
    ON 0000 NEXT
  FIND CURRENT JOB
    ON 0000 NEXT
  MODIFY JOB
    ON 0000 NEXT
.
ADD
PATH-GROUP NAME IS OBTAIN EMP-EMPOSITION
SELECT
  OBTAIN FIRST EMPLOYEE ←----- CALC
    WHERE CALCKEY EQ EMP-ID-0415      entry
    OF REQUEST
    ON 0000 NEXT
    ON 0326 RETURN INVALID-ID
  OBTAIN EACH EMPOSITION WITHIN EMP-EMPOSITION
    ON 0000 NEXT
    ON 0307 ITERATE
  OBTAIN OWNER WITHIN JOB-EMPOSITION
    ON 0000 NEXT
.
```

## Example 2—Adding Path-groups to Expand Functionality

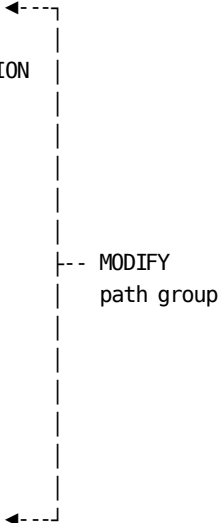
This example shows how to add a STORE and ERASE path group to the ASF-generated table definition described above.

### ASF-generated Subschema

```

ADD
SUBSCHEMA NAME IS RU000199 OF SCHEMA NAME IS IDMSR VERSION IS 1
.
.
.
ADD
LOGICAL RECORD NAME IS EMP-EMPOSITION
ELEMENTS ARE
    ASF-RDEF-REC
    EMPLOYEE
    EMPOSITION
    JOB
.
ADD
PATH-GROUP NAME IS MODIFY EMP-EMPOSITION
SELECT
    FIND CURRENT EMPLOYEE
        ON 0000 NEXT
    MODIFY EMPLOYEE
        ON 0000 NEXT
    FIND CURRENT EMPOSITION
        ON 0000 NEXT
    MODIFY EMPOSITION
        ON 0000 NEXT
    FIND CURRENT JOB
        ON 0000 NEXT
    MODIFY JOB
        ON 0000 NEXT
.

```



MODIFY  
path group

```
ADD
PATH-GROUP NAME IS OBTAIN EMP-EMPOSITION
SELECT
  FIND FIRST ASF-RDEF-OWN WITHIN IDMSR-AREA
  ON 0000 NEXT
  ON 0307 ITERATE
  OBTAIN EACH EMPLOYEE WITHIN EMP-DEMO-REGION
  ON 0000 NEXT
  ON 0307 ITERATE
  OBTAIN EACH EMPOSITION WITHIN EMP-EMPOSITION
  ON 0000 NEXT
  ON 0307 ITERATE
  OBTAIN OWNER WITHIN JOB-EMPOSITION
  ON 0000 NEXT
.
```

--- OBTAIN  
path group

**Modified Subschema**

```
ADD
SUBSCHEMA NAME IS RU000199 OF SCHEMA NAME IS IDMSR VERSION IS 1
.
.
.
ADD
LOGICAL RECORD NAME IS EMP-EMPOSITION
ELEMENTS ARE
  ASF-RDEF-REC
  EMPLOYEE
  EMPOSITION
  JOB
```

```
ADD
PATH-GROUP NAME IS ERASE EMP-EMPOSITION ←----- New ERASE
SELECT                                          path group
  FIND CURRENT EMPLOYEE
    ON 0000 NEXT
    ON 0306 DO
      OBTAIN FIRST EMPLOYEE
        WHERE CALCKEY EQ EMP-ID-0415
          OF LR
            ON 0000 NEXT
            ON 0326 RETURN NO-EMPLOYEE
      END
    OBTAIN EACH EMPOSITION WITHIN EMP-EMPOSITION
      ON 0000 NEXT
      ON 0307 ITERATE
    ERASE EMPOSITION
      ON 0000 ITERATE
.
```

```
ADD
PATH-GROUP NAME IS MODIFY EMP-EMPOSITION
SELECT
  FIND CURRENT EMPLOYEE
    ON 0000 NEXT
  MODIFY EMPLOYEE
    ON 0000 NEXT
  FIND CURRENT EMPOSITION
    ON 0000 NEXT
  MODIFY EMPOSITION
    ON 0000 NEXT
  FIND CURRENT JOB
    ON 0000 NEXT
  MODIFY JOB
    ON 0000 NEXT
.
```

```
ADD
PATH-GROUP NAME IS OBTAIN EMP-EMPOSITION
SELECT
  FIND FIRST ASF-RDEF-OWN WITHIN IDMSR-AREA
  ON 0000 NEXT
  ON 0307 ITERATE
  OBTAIN EACH EMPLOYEE WITHIN EMP-DEMO-REGION
  ON 0000 NEXT
  ON 0307 ITERATE
  OBTAIN EACH EMPOSITION WITHIN EMP-EMPOSITION
  ON 0000 NEXT
  ON 0307 ITERATE
  OBTAIN OWNER WITHIN JOB-EMPOSITION
  ON 0000 NEXT
.
```

```
ADD
PATH-GROUP NAME IS STORE EMP-EMPOSITION ←----- New STORE
SELECT                                          path group
  FIND CURRENT EMPLOYEE
  ON 0000 NEXT
  ON 0306 DO
    OBTAIN FIRST EMPLOYEE
    WHERE CALCKEY EQ EMP-ID-0415
    OF EMPLOYEE OF LR
    ON 0000 NEXT
    ON 0326 RETURN NO-EMPLOYEE
  END
  STORE EMPOSITION
  ON 0000 NEXT
.
```



## Example 3—Implementing Referential Integrity Checks for a Table

This example shows how to check for a valid employee ID before a new EDUCATION row is stored.

### Screen 1

Column Name	Width	Type	Unique Key	Display Seq	Ext Screen
EMP-ID	4	NUMERIC	X	1	-
COURSE-ID	4	NUMERIC	-	2	-
COURSE-NAME	20	TEXT	-	3	-

### ASF-generated Subschema

```

ADD
SUBSCHEMA NAME IS RU000189 OF SCHEMA NAME IS IDMSR VERSION IS 1
.
.
.
ADD
LOGICAL RECORD NAME IS EDUCATION
ELEMENTS ARE
  ASF-RDEF-REC
  RFUR-000189-00AK
  RFUR-000189-DATA
.
.
.
ADD
PATH-GROUP NAME IS STORE EDUCATION
SELECT
  FIND CURRENT RFUR-000189-DATA
  ON 0306 DO
  FIND FIRST ASF-RDEF-OWN WITHIN IDMSR-AREA
  ON 0000 NEXT
  ON 0307 ITERATE
  OBTAIN FIRST ASF-RDEF-REC WITHIN RDEFSET
  WHERE SORTKEY EQ 000000189
  ON 0326 RETURN DEFN-MISSING
  ON 0000 NEXT

```

```
OBTAIN RFUR-000189-00AK
  WHERE DBKEY EQ RDEF-OWNER-DBKEY OF LR
  ON 0326 RETURN 00AK-MISSING
  ON 0000 NEXT
FIND LAST RFUR-000189-DATA WITHIN RFPS-000189-SET
  ON 0307 NEXT
  ON 0000 NEXT
  END
  ON 0000 NEXT
STORE RFUR-000189-DATA
  ON 0000 NEXT
```

.  
.
.  
.

### Modified Subschema

```
ADD
SUBSCHEMA NAME IS RJ000189 OF SCHEMA NAME IS IDMSR VERSION IS 1
```

.  
.

```
ADD
LOGICAL RECORD NAME IS EDUCATION
ELEMENTS ARE
  ASF-RDEF-REC
  RFUR-000189-00AK
  RFUR-000189-DATA
```

.  
.

```
ADD
PATH-GROUP NAME IS STORE EDUCATION
SELECT
```

```
FIND FIRST EMPLOYEE
  WHERE CALCKEY EQ EMP-ID OF LR
  ON 0326 RETURN INVALID-ID
  ON 0000 NEXT
```

←---- Checks for a valid employee ID

```
FIND CURRENT RFUR-000189-DATA
  ON 0306 DO
FIND FIRST ASF-RDEF-OWN WITHIN IDMSR-AREA
  ON 0000 NEXT
  ON 0307 ITERATE
OBTAIN FIRST ASF-RDEF-REC WITHIN RDEFSET
  WHERE SORTKEY EQ 000000189
  ON 0326 RETURN DEFN-MISSING
  ON 0000 NEXT
OBTAIN RFUR-000189-00AK
  WHERE DBKEY EQ RDEF-OWNER-DBKEY OF LR
  ON 0326 RETURN 00AK-MISSING
  ON 0000 NEXT
FIND LAST RFUR-000189-DATA WITHIN RFPS-000189-SET
  ON 0307 NEXT
  ON 0000 NEXT
  END
  ON 0000 NEXT
STORE RFUR-000189-DATA
  ON 0000 NEXT
```

.  
.

## Example 4—Performing a Difference Join

This example shows how to perform a difference join for the CUSTOMER and ORDER tables, which retrieves all customers who do not have orders.

To perform this join, you first join the CUSTOMER and ORDER tables through ASF, specifying the CUSTOMER table as source table #1. Then you modify the ASF-generated subschema as shown below.

### Screen 1

```

CA
CDEF CA - Automatic System Facility      ** Column Definition **
DC560302 ADD/MODIFY COLUMNS
Table Name: CUSTOMER

  _ PF1 - Help                          _ PF3 - Extended Selection Definition
  _ PF2 - Extended Column Definition    _ PF4 - Extended Key Definition

Page: 1 of 1

```

Column Name	Width	Type	Unique Key	Display Seq	Ext Screen
CUSTNUM	4	NUMERIC	X	1	-
CUSTNAME	30	TEXT	-	2	-
STREET	30	TEXT	-	3	-
CITY	30	TEXT	-	4	-
ZIP	9	TEXT	-	5	-
SALESREP	4	NUMERIC	-	6	-

### Screen 2

```

CA
CDEF CA - Automatic System Facility      ** Column Definition **
DC560302 ADD/MODIFY COLUMNS
Table Name: ORDER

  _ PF1 - Help                          _ PF3 - Extended Selection Definition
  _ PF2 - Extended Column Definition    _ PF4 - Extended Key Definition

Page: 1 of 1

```

Column Name	Width	Type	Unique Key	Display Seq	Ext Screen
CUSTNUM	4	NUMERIC	-	1	-
ORDNUM	4	NUMERIC	X	2	-
AMOUNT	6	CURRENCY	-	3	-

**Screen 3**

```

CA
TDEF CA - Automatic System Facility          ** Table Definition **
DC560005 MODIFY AND/OR SELECT NEXT ACTIVITY

    _ PF1 - Help                          _ PF4 - Extended Table Definition
    _ PF2 - Define Columns                 _ PF5 - Delete Table Definition
    _ PF3 - Generate                       _ PF6 - Message Screen

Table Name.: CHEAP-CUSTOMERS
Table Owner: JSS                          Defn Number: 106
View/Stored: VIEW                          Status.....: GENERATED
Comments...:

                                Table Derivation
Source Table #1
Table Name.: CUSTOMER
Table Owner: JSS

Source Table #2
Table Name.: ORDER
Table Owner: JSS

Column #1:
Column #2                                Where Column #1 EQ Column #2

```

**ASF-generated Subschema (Regular Join)**

```

ADD
SUBSCHEMA NAME IS RU000106 OF SCHEMA NAME IS IDMSR VERSION IS 1
.
.
ADD
LOGICAL RECORD NAME IS CHEAP-CUSTOMERS
ELEMENTS ARE
ASF-RDEF-REC
RFUR-000101-00AK
RFUR-000102-00AK
RFUR-000101-DATA
RFUR-000102-DATA
.
.

```

```

ADD
PATH-GROUP NAME IS OBTAIN CHEAP-CUSTOMERS
.
.
.
  FIND FIRST ASF-RDEF-OWN WITHIN IDMSR-AREA
    ON 0000 NEXT
    ON 0307 ITERATE
  OBTAIN FIRST ASF-RDEF-REC WITHIN RDEFSET
    WHERE SORTKEY EQ 000000101
    ON 0326 RETURN DEFN-MISSING
    ON 0000 NEXT

OBTAIN RFUR-000101-00AK
  WHERE DBKEY EQ RDEF-OWNER-DBKEY OF ASF-RDEF-REC OF LR
  ON 0326 RETURN 00AK-MISSING
  ON 0000 NEXT
EVALUATE RDEF-TIME-STAMP
  OF ASF-RDEF-REC OF LR LE OWNER-TIME-STAMP
  OF RFUR-000101-00AK OF LR
  ON 2001 RETURN SYNC-ERROR
  ON 0000 NEXT
OBTAIN FIRST ASF-RDEF-REC WITHIN RDEFSET
  WHERE SORTKEY EQ 000000102
  ON 0326 RETURN DEFN-MISSING
  ON 0000 NEXT
OBTAIN RFUR-000102-00AK
  WHERE DBKEY EQ RDEF-OWNER-DBKEY OF ASF-RDEF-REC OF LR
  ON 0326 RETURN 00AK-MISSING
  ON 0000 NEXT
EVALUATE RDEF-TIME-STAMP OF ASF-RDEF-REC
  OF LR LE OWNER-TIME-STAMP
  OF RFUR-000102-00AK OF LR
  ON 2001 RETURN SYNC-ERROR
  ON 0000 NEXT
OBTAIN EACH RFUR-000101-DATA USING INDEX
  ON 0326 ITERATE
  ON 0000 NEXT
OBTAIN EACH RFUR-000102-DATA USING RFFS-000102-0001
  WHERE CUSTNUM OF RFUR-000102-DATA EQ CUSTNUM OF
    RFUR-000101-DATA OF LR
  ON 0326 ITERATE ←----- Path iterates when a
  ON 0000 NEXT      match on CUSTNUM is not found.

```

### Modified Subschema (Difference Join)

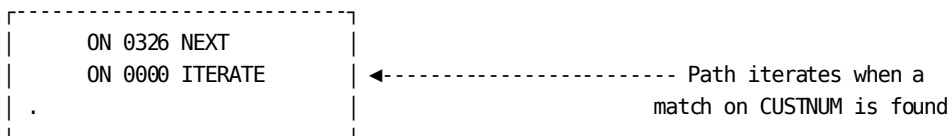
```
ADD
SUBSCHEMA NAME IS RU000106 OF SCHEMA NAME IS IDMSR VERSION IS 1
.
.
ADD
LOGICAL RECORD NAME IS CHEAP-CUSTOMERS
ELEMENTS ARE
    ASF-RDEF-REC
    RFUR-000101-00AK
    RFUR-000102-00AK
    RFUR-000101-DATA
    RFUR-000102-DATA
.
ADD
PATH-GROUP NAME IS OBTAIN CHEAP-CUSTOMERS
.
.
FIND FIRST ASF-RDEF-OWN WITHIN IDMSR-AREA
    ON 0000 NEXT
    ON 0307 ITERATE
OBTAIN FIRST ASF-RDEF-REC WITHIN RDEFSET
    WHERE SORTKEY EQ 000000101
    ON 0326 RETURN DEFN-MISSING
    ON 0000 NEXT
OBTAIN RFUR-000101-00AK
    WHERE DBKEY EQ RDEF-OWNER-DBKEY OF ASF-RDEF-REC OF LR
    ON 0326 RETURN 00AK-MISSING
    ON 0000 NEXT

EVALUATE RDEF-TIME-STAMP OF ASF-RDEF-REC
    OF LR LE OWNER-TIME-STAMP
    OF RFUR-000101-00AK OF LR
    ON 2001 RETURN SYNC-ERROR
    ON 0000 NEXT
OBTAIN FIRST ASF-RDEF-REC WITHIN RDEFSET
    WHERE SORTKEY EQ 000000102
    ON 0326 RETURN DEFN-MISSING
    ON 0000 NEXT
OBTAIN RFUR-000102-00AK
    WHERE DBKEY EQ RDEF-OWNER-DBKEY OF ASF-RDEF-REC OF LR
    ON 0326 RETURN 00AK-MISSING
```

```

ON 0000 NEXT
EVALUATE RDEF-TIME-STAMP OF ASF-RDEF-REC
  OF LR LE OWNER-TIME-STAMP
  OF RFUR-000102-00AK OF LR
ON 2001 RETURN SYNC-ERROR
ON 0000 NEXT
OBTAIN EACH RFUR-000101-DATA USING INDEX
ON 0326 ITERATE
ON 0000 NEXT
OBTAIN EACH RFUR-000102-DATA USING RFFS-000102-0001
  WHERE CUSTNUM OF RFUR-000102-DATA EQ CUSTNUM OF
    RFUR-000101-DATA OF LR

```



## Example 5—Calculating Summary Information

This example shows how to calculate summary information for the CUSTOMER and ORDER tables. This information includes a count of the orders for each customer and a total of each customer's order amounts.

To calculate summary information, you can first join the tables through ASF. When you join the tables, you should add two IDD-defined work fields (one will contain the order total and the other will contain a count of orders). You should also add a calculation for each field. The COLUMN(AMOUNT) calculation provides a good start.

**Screen 1**

```

CA
TDEF CA - Automatic System Facility          ** Table Definition **
DC560005 MODIFY AND/OR SELECT NEXT ACTIVITY

    _ PF1 - Help                          _ PF4 - Extended Table Definition
    _ PF2 - Define Columns                 _ PF5 - Delete Table Definition
    _ PF3 - Generate                       _ PF6 - Message Screen

Table Name.: CUST-ORD-SUMMARY
Table Owner: JSS                          Defn Number: 105
View/Stored: VIEW                          Status.....: GENERATED
Comments...:

                                Table Derivation
Source Table #1
Table Name.: CUSTOMER
Table Owner: JSS

Source Table #2
Table Name.: ORDER
Table Owner: JSS

Column #1:
Column #2                                Where Column #1 EQ Column #2
    
```

**Screen 2**

```

CA
XCOL CA - Automatic System Facility          *Extended Column Definition*
DC560407 MODIFY COLUMN DEFINITION
Table Name: CUST-ORD-SUMMARY
    _ PF1 - Help          _ PF2 - Next Column          _ PF3 - Message Screen

    Displayed Column Name          Stored Column Name
    CUST-TOTAL                    CUST-TOTAL

    Display Format          Stored Format
    -Z(16).9(02)          S9(16)V9(02)

Usage: 3 D/0/1/2/3 Display/Comp/Comp-1/Comp-2/Comp-3

Comments.....:          Column Level: WORK
Initial Value:
Calculation...: COLUMN(AMOUNT)

Edit Table:          Version:          Code Table:          Version:

U U/P Unprotected/Protected N B/R/P/T/G/Y/W/N Blue/Red/Pink/Turquoise
Green/Yellow/White/NoColor

r
N Y/N Blank When Zero          N N/R Normal/Reverse Video

D B/D/K Bright/Display/Dark          N Y/N Blink/NoBlink
    
```



## Screen 3

```

                                CA
XCOL  CA - Automatic System Facility      *Extended Column Definition*
DC560407 MODIFY COLUMN DEFINITION
Table Name: CUST-ORD-SUMMARY
      _ PF1 - Help      _ PF2 - Next Column      _ PF3 - Message Screen

      Displayed Column Name      Stored Column Name
ORDER-COUNT                      ORDER-COUNT

      Display Format      Stored Format
-Z(16).9(02)            S9(16)V9(02)

Usage: 3 D/0/1/2/3  Display/Comp/Comp-1/Comp-2/Comp-3

Comments.....:      Column Level: WORK
Initial Value:
Calculation..: COLUMN(AMOUNT)

Edit Table:      Version:      Code Table:      Version:

  U  U/P  Unprotected/Protected  N  B/R/P/T/G/Y/W/N  Blue/Red/Pink/Turquoise
Green/Yellow/White/Nocolor
  N  Y/N  Blank When Zero      N  N/R  Normal/Reverse Video

  D  B/D/K  Bright/Display/Dark  N  Y/N  Blink/NoBlink

```

## ASF-defined Subschema

```

ADD
SUBSCHEMA NAME IS RU000105 OF SCHEMA NAME IS IDMSR VERSION IS 1
.
.
.
ADD
LOGICAL RECORD NAME IS CUST-ORD-SUMMARY
ELEMENTS ARE
  ASF-RDEF-REC
  RFUR-000101-00AK
  RFUR-000102-00AK
  RFUR-000101-DATA
  RFUR-000102-DATA
  RFUR-000105-WORK VERSION 1
.

```

```

ADD
PATH-GROUP NAME IS OBTAIN CUST-ORD-SUMMARY
.
.
.
FIND FIRST ASF-RDEF-OWN WITHIN IDMSR-AREA
  ON 0000 NEXT
  ON 0307 ITERATE
OBTAIN FIRST ASF-RDEF-REC WITHIN RDEFSET
  WHERE SORTKEY EQ 000000101
  ON 0326 RETURN DEFN-MISSING
  ON 0000 NEXT
OBTAIN RFUR-000101-00AK
  WHERE DBKEY EQ RDEF-OWNER-DBKEY OF ASF-RDEF-REC OF LR
  ON 0326 RETURN 00AK-MISSING
  ON 0000 NEXT
EVALUATE RDEF-TIME-STAMP OF ASF-RDEF-REC
  OF LR LE OWNER-TIME-STAMP
  OF RFUR-000101-00AK OF LR
  ON 2001 RETURN SYNC-ERROR
  ON 0000 NEXT
OBTAIN FIRST ASF-RDEF-REC WITHIN RDEFSET
  WHERE SORTKEY EQ 000000102
  ON 0326 RETURN DEFN-MISSING
  ON 0000 NEXT
OBTAIN RFUR-000102-00AK
  WHERE DBKEY EQ RDEF-OWNER-DBKEY OF ASF-RDEF-REC OF LR
  ON 0326 RETURN 00AK-MISSING
  ON 0000 NEXT
EVALUATE RDEF-TIME-STAMP OF ASF-RDEF-REC
  OF LR LE OWNER-TIME-STAMP
  OF RFUR-000102-00AK OF LR
  ON 2001 RETURN SYNC-ERROR
  ON 0000 NEXT
OBTAIN EACH RFUR-000101-DATA USING INDEX
  ON 0326 ITERATE
  ON 0000 NEXT
OBTAIN EACH RFUR-000102-DATA USING RFFS-000102-0001
  WHERE CUSTNUM OF RFUR-000102-DATA EQ CUSTNUM OF
    RFUR-000101-DATA OF LR
  ON 0326 ITERATE
  ON 0000 NEXT
COMPUTE CUST-TOTAL OF RFUR-000105-WORK OF LR = AMOUNT OF ←-}
  RFUR-000102-DATA OF LR                                     }--- Work field
  ON 0000 NEXT                                               |   calculations
COMPUTE ORDER-COUNT OF RFUR-000105-WORK OF LR = AMOUNT OF ←-}
  RFUR-000102-DATA OF LR
  ON 0000 NEXT
.

```

**Modified Subschema**

```
ADD
SUBSCHEMA NAME IS RU000105 OF SCHEMA NAME IS IDMSR VERSION IS 1
.
.
.
ADD
LOGICAL RECORD NAME IS CUST-ORD-SUMMARY
ELEMENTS ARE
    ASF-RDEF-REC
    RFUR-000101-00AK
    RFUR-000102-00AK
    RFUR-000101-DATA
    RFUR-000102-DATA
    RFUR-000105-WORK VERSION 1
.
ADD
PATH-GROUP NAME IS OBTAIN CUST-ORD-SUMMARY
.
.
.
FIND FIRST ASF-RDEF-OWN WITHIN IDMSR-AREA
    ON 0000 NEXT
    ON 0307 ITERATE
OBTAIN FIRST ASF-RDEF-REC WITHIN RDEFSET
    WHERE SORTKEY EQ 000000101
    ON 0326 RETURN DEFN-MISSING
    ON 0000 NEXT
OBTAIN RFUR-000101-00AK
    WHERE DBKEY EQ RDEF-OWNER-DBKEY OF ASF-RDEF-REC OF LR
    ON 0326 RETURN 00AK-MISSING
    ON 0000 NEXT
EVALUATE RDEF-TIME-STAMP OF ASF-RDEF-REC
    OF LR LE OWNER-TIME-STAMP
    OF RFUR-000101-00AK OF LR
    ON 2001 RETURN SYNC-ERROR
    ON 0000 NEXT
OBTAIN FIRST ASF-RDEF-REC WITHIN RDEFSET
    WHERE SORTKEY EQ 000000102
    ON 0326 RETURN DEFN-MISSING
    ON 0000 NEXT
OBTAIN RFUR-000102-00AK
    WHERE DBKEY EQ RDEF-OWNER-DBKEY OF ASF-RDEF-REC OF LR
    ON 0326 RETURN 00AK-MISSING
    ON 0000 NEXT
```

```

EVALUATE RDEF-TIME-STAMP OF ASF-RDEF-REC
  OF LR LE OWNER-TIME-STAMP
  OF RFUR-000102-00AK OF LR
  ON 2001 RETURN SYNC-ERROR
  ON 0000 NEXT
OBTAIN EACH RFUR-000101-DATA USING INDEX
  ON 0326 ITERATE

```

```

-----
| ON 0000 DO |
| COMPUTE CUST-TOTAL OF RFUR-000105-WORK OF LR = 0 <-- |
| ON 0000 NEXT | }-- Initializes |
| COMPUTE ORDER-COUNT OF RFUR-000105-WORK OF LR = 0 <-- | the work fields |
| ON 0000 NEXT | |
| END |
| OBTAIN EACH RFUR-000102-DATA USING RFFS-000102-0001 |
| WHERE CUSTNUM OF RFUR-000102-DATA EQ CUSTNUM OF |
| RFUR-000101-DATA OF LR |
| ON 0326 RETURN LR-FOUND <----- Returns order amounts |
| ON 0000 DO | and counts per customer |
| COMPUTE CUST-TOTAL OF RFUR-000105-WORK OF LR = CUST-TOTAL OF <-- Totals the |
| RFUR-000105-WORK OF LR + AMOUNT OF RFUR-000102-DATA OF LR order amounts |
| ON 0000 NEXT | per customer |
| COMPUTE ORDER-COUNT OF RFUR-000105-WORK OF LR = ORDER-COUNT OF <-- Counts the |
| RFUR-000105-WORK OF LR + 1 | number of |
| ON 0000 ITERATE <----- | orders per |
| END | customer |
| . |
| ----- |

```

# Appendix B: Performing UNLOAD/RELOAD Against an ASF Database

---

ASF data and definition areas can be processed by the UNLOAD and RELOAD utilities but additional steps must be taken to insure that the operation is successful.

This section contains the following topics:

[Procedure](#) (see page 341)

[Preparing Your Environment](#) (see page 342)

## Procedure

Running an UNLOAD/RELOAD against an area associated with an ASF database requires the following steps:

1. Prepare the environment to run the UNLOAD/RELOAD utilities
2. Run the IDMSRSSM program to:
  - Create a syntax for the IDMSRSSD subschema that you will use to unload and reload the ASF data areas
3. Review the IDMSRSSD subschema syntax produced by the IDMSRSSM program and generate it.
4. UNLOAD and RELOAD the ASF data or definition areas.
5. Run the IDMSRUPD program to update the RDEFREC records in the ASF definition area with the new db-keys of the table header records (RFUR-nnnnnn-OOAK).

Each of these steps is described in the remainder of this appendix.

## Preparing Your Environment

To prepare your environment implement the following steps:

1. Identify the areas and the ASF-defined tables you want to unload and reload. Make sure that the tables you will process are completely generated. A table that is not completely generated or does not have a validated subschema will not be included in the unload/reload.
2. The IDMSRSSM program will identify all tables which cannot be processed. If you want to include such a table, you must first generate or re-generate the table using ASF.
3. Run the PRINT SPACE utility against each ASF data and definition area you will unload/reload. You can compare the records listed on the PRINT SPACE report with the output from the IDMSRSSM program to determine if all tables are processed by IDMSRSSM.
4. Backup your ASF dictionary and areas.
5. Run the IDMSRSSM program.

## Running the IDMSRSSM Program

Once the environment is prepared, run the IDMSRSSM program against the ASF dictionary. The IDMSRSSM program:

- Creates the subschema syntax for the IDMSRSSD subschema and places it in an output file. The IDMSRSSD subschema is used to unload and reload ASF data areas.
- Produces a listing of record ID's for the tables in the IDMSR schema version 1.

:Caution.

Tables that don't have an existing validated subschema are not included in the IDMSRSSD subschema.

You can run the IDMSRSSM program against the same ASF dictionary as often as necessary.

### Syntax

▶▶ AREA=*asf-data-area-name* ▶▶▶▶

**Parameter** AREA=*asf-data-area-name*

Specifies the name of the ASF data area you wish to unload and reload.  
'asf-data-area-name' must be in an ASF data area defined in the dictionary.

You can specify a maximum of 20 area statements for each run of the IDMSRSSM program. If the unload/reload is to be performed against the ASF definition area the IDMSRSSM parameters should include all ASF data areas that have table definitions in the target ASF definition area.

## IDMSRSSM Execution JCL

IDMSRSSM is a program in the load library provided at installation.

IDMSRSSM runs in a batch environment in either local mode or under the central version.

Execution JCL for the IDMSRSSM program is provided below.

### z/OS JCL

#### IDMSRSSM

```
//BUILD EXEC PGM=IDMSRSSM,REGION=
//STEPLIB DD DSN=idms.dba.loadlib,DISP=SHR
// DD DSN=idms.custom.loadlib,DISP=SHR
// DD DSN=idms.cagjload,DISP=SHR
//asfdict DD DSN=idms.asfdict.ddldml,DISP=SHR
//dcmmsg DD DSN=idms.sysmsg.ddldcmmsg,DISP=SHR
//sysjrnl DD DSN=idms.tapejrnl,DISP=(NEW,KEEP),UNIT=TAPE
//SYS004 DD DSN=output.file,DISP=(,CATLG),UNIT=
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=nnnn)
//SYSLST DD SYSOUT=A
//SYSIDMS DD *
```

```
DMCL=dmcl-name
DBNAME=dbname or segment-name
```

```
//SYSIPT DD *
```

```
AREA=ASF-data-area-name AREA=ASF-data-area-name
```

<i>idms.dba.loadlib</i>	Data set name of the load library containing the DMCL and database name table load modules
<i>idms.custom.loadlib</i>	Data set name of the load library containing customized CA IDMS load modules

<i>idms.cagjload</i>	Data set name of the load library containing executable CA IDMS modules that do not require customization
<i>asfdict</i>	DDname of the ASF (DDLML) dictionary area
<i>idms.asfdict.ddldml</i>	Data set name of the ASF (DDLML) dictionary area
<i>dcmsg</i>	DDname of the system message (DDLDCMSG) area
<i>idms.sysmsg.ddldcmsg</i>	Data set name of the system message (DDLDCMSG) area
<i>sysjml</i>	DDname of a tape journal file, if journaling
<i>idms.tapejml</i>	Dataset name of tape journal file
<i>sys004</i>	DDname of the output file for the subschema produced by the IDMSRSSM program
<i>output.file</i>	Data set name of the output file for the subschema produced by the IDMSRSSM program
<i>dmcl-name</i>	The name of a DMCL
<i>dbname or segment-name</i>	Database name or segment name identifying the DDLML area of the ASF dictionary
<i>ASF-data-area-name</i>	The name of an ASF extent area

To run IDMSRSSM under the central version, add a SYSCTL file.

**z/VSE JCL IDMSRSSM**

```
// JOB IDMSRSSM
// LIBDEF *,SEARCH=idms libraries
// EXEC PROC=IDMSLBLE
// ASSGN sysnnn,DISK,VOL=nnnnnn,SHR
// DLBL SYS004,'output.file'
// EXEC IDMSRSSM,SIZE=1048K
DMCL=dmcl-name
DBNAME=dbname or segment-name
/*
AREA=ASF-data-area-name AREA=ASF-data-area-name
/*
```

<i>IDMSLBLE</i>	Name of the procedure provided at installation that contains the file definitions for CA IDMS dictionaries, databases, disk journal files, and the SYSIDMS file. For a complete listing of IDMSLBLE, see the <i>CA IDMS Conversion Guide</i> .
<i>sys004</i>	Filename of the output file for the subschema produced by IDMSRSSM program



<i>dmcl-name</i>	The name of a DMCL
<i>dbname or segment-name</i>	Database name or segment name identifying the DDLML area of the ASF dictionary
<i>ASF-data-area-name</i>	The name of an ASF extent area

To run IDMSRSSM under the central version, add a SYSCTL file.

### VM/ESA Commands

#### IDMSRSSM

```
FILEDEF SYSLST PRINTER
FI asfdict DISK cdms asfdict fm (RECFM F LRECL ppp BLKSIZE ppp XTENT nnn
FI dcmsg DISK cdms dmsgdb fm (RECFM F LRECL ppp BLKSIZE ppp XTENT nnn
FILEDEF sysjml TAP1 SL VOLID nnnnn (RECFM VB LRECL lll BLKSIZE bbbb
FILEDEF SYS004 DISK sys004 output a
FILEDEF SYSIDMS DISK sysidms input a
FILEDEF SYSIPT DISK idmsrsm input a
GLOBAL LOADLIB dbalib idmslib
OSRUN IDMSRSSM
```

<i>asfdict</i>	DDname of the ASF (DDLML) dictionary area
<i>cdms asfdict fm</i>	File identifier of the ASF (DDLML) dictionary area
<i>dcmsg</i>	DDname of the system message (DDLDCMSG) area
<i>cdms dmsgdb fm</i>	File identifier of the system message (DDLDCMSG) area
<i>ppp</i>	Page size of the database file
<i>nnn</i>	Number of pages in the database file
<i>sysjml</i>	DDname of the tape journal file, if journaling
<i>nnnnn</i>	Volume serial number of the tape journal file
<i>lll</i>	Record length of the tape journal file
<i>bbbb</i>	Block size of the tape journal file
<i>sys004 output a</i>	File identifier of file containing subschema produced by IDMSRSSM program
<i>sysidms input a</i>	File identifier of the file containing the following: <ul style="list-style-type: none"> <li>■ <i>dmcl-name</i>— Name of the DMCL</li> <li>■ <i>segment-name</i>— database name or segment name identifying the DDLML area of the ASF dictionary</li> </ul>
<i>idmsrsm input a</i>	File identifier of the file containing the names of the ASF extent areas you will migrate

<i>dbalib</i>	Filename of the load library containing the DMCL and database name table load modules
<i>idmslib</i>	Filename of the load library containing executable CA IDMS modules

## Output from IDMSRSSM

### What IDMSRSSM Produces

- An output listing that includes various information which is described in the table below.
- A subschema syntax for the IDMSRSSD subschema.

### IDMSRSSM Output Listing Sample

Portions of sample output listing from IDMSRSSM are provided below. Allocation limits are represented as tables instead of K bytes and allocation defaults are removed from the catalog.

```

RECORD ID: 31839
RECORD ID: 31840
RECORD ID: 31841
RECORD ID: 31842
VALIDATE NOT DONE FOR SUBSCHEMA:
RECORD: RFUR-000217-00AK          WILL NOT BE INCLUDED
IN SYNTAX FOR AREA: IDMSR-AREA2

RECORD ID: 31913
VALIDATE NOT DONE FOR SUBSCHEMA:
RECORD: RFUR-000217-DATA          WILL NOT BE INCLUDED
IN SYNTAX FOR AREA: IDMSR-AREA2

RECORD ID: 31914
VALIDATE NOT DONE FOR SUBSCHEMA:
RECORD: RFUR-000218-00AK          WILL NOT BE INCLUDED
IN SYNTAX FOR AREA: IDMSR-AREA2
RECORD ID: 31934
RECORD: RFOR-000135-00AK          WILL NOT BE INCLUDED
IN SYNTAX FOR AREA: IDMSR-AREA2

RECORD ID: 31935
RECORD: RFOR-000135-DATA          WILL NOT BE INCLUDED
IN SYNTAX FOR AREA: IDMSR-AREA2

RECORD ID: 31936
RECORD ID: 31998
RECORD ID: 31999
AREAS ADDED: 0002 RECORDS ADDED: 0140 SETS ADDED: 0124

                UPDATING CATALOG

USER: CORP                                EXG
VALUES FOUND: TOTAL USED: 00002291      COUNT OF BYTES: 00002291
REPLACED WITH TABLE COUNT: 00010
COUNT OF TABLES UNALLOCATED: 00001

USER: CORP                                W FURR
VALUES FOUND: TOTAL USED: 00000000      COUNT OF BYTES: 00000000
REPLACED WITH TABLE COUNT: 00000

USER: CORP                                DNT
VALUES FOUND: TOTAL USED: 00001014      COUNT OF BYTES: 00001014
REPLACED WITH TABLE COUNT: 00004
COUNT OF TABLES UNALLOCATED: 00004

USER: CORP                                SJG
USER DEFAULTS TOTAL ALLOCATION: 00000100
BEING REMOVED - PRIMARY: 00000 SECONDARY: 00000 MAXIMUM: 00000
VALUES FOUND: TOTAL USED: 00030468      COUNT OF BYTES: 00030968
REPLACED WITH TABLE COUNT: 00011

USER: CORP                                CULL DBA
VALUES FOUND: TOTAL USED: 00001512      COUNT OF BYTES: 00001513
REPLACED WITH TABLE COUNT: 00002
COUNT OF TABLES UNALLOCATED: 00001

COUNT OF USERS UPDATED: 00025

```

For more information, see the *CA IDMS Conversion Guide*.

## Review the IDMSRSSM Output Listing

Much of the information produced on the output listing only had meaning when migrating of the ASF areas to a post-10.2 environment. The following table identifies those items of interest when performing an UNLOAD/RELOAD after an earlier migration has completed.

Item in report	Item meaning	What to do
RECORD ID	The record id of a table included in the IDMSRSSD subschema	If you produced a PRINT SPACE report, compare it with the list of record ids produced by IDMSRSSM to determine if all tables were processed.
VALIDATE NOT DONE FOR SUBSCHEMA	Identifies the record ID of a table that does not have a validated subschema.	If you want to unload this table, use the ASF to determine why the subschema is not validated, fix the problem and generate the table.
Record: RFOR-nnnnnn-DATA WILL NOT BE INCLUDED IN SYNTAX	Identifies tables that have existing definition record names preceded with 'RFOR-'. These names indicated that a table was not completely generated. These records are not included in the IDMSRSSD subschema.	If you want to unload the data from tables that correspond to an existing 'RFOR' record, access the table through ASF and complete the regeneration process.

## Generate the IDMSRSSD Subschema

After reviewing the IDMSRSSM output, use the subschema compiler to generate the IDMSRSSD subschema.

**Note:** IDMSRSSD is the default subschema name used by the IDMSRUPD program when an ASF definition is processed by the UNLOAD/RELOAD utilities and a subschema name is not provided. You can change the subschema name before generating it. If you do, be sure to specify the new name when running the UNLOAD and RELOAD utility against ASF data areas and the IDMSRUPD program.



### Parameters

#### SUB=Subschema-name

Specifies the name of the subschema the IDMSRUPD program will use. 'subschema-name' must be a 1 through 8 character alphanumeric value.

If you do not specify a 'subschema-name' IDMSRUPD will use IDMSRSSD as the default subschema name.

## IDMSRUPD Execution JCL

IDMSRSSM is a program in the load library provided at installation.

IDMSRSSM runs in a batch environment in either local mode or under the central version.

Execution JCL for the IDMSRSSM program is provided below.

### z/OS JCL

#### IDMSRSSM

```
//BUILD EXEC PGM=IDMSRSSM,REGION=
//STEPLIB DD DSN=idms.dba.loadlib,DISP=SHR
// DD DSN=idms.custom.loadlib,DISP=SHR
// DD DSN=idms.cagjload.,DISP=SHR
//asfdict DD DSN=idms.asfdict.ddldml,DISP=SHR
//dcmmsg DD DSN=idms.sysmsg.ddldcmmsg,DISP=SHR
//sysjrnl DD DSN=idms.tapejrnl,DISP=(NEW,KEEP),UNIT=TAPE
//SYS004 DD DSN=output.file,DISP=(,CATLG),UNIT=
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=nnnn)
//SYSLST DD SYSOUT=A
//SYSIDMS DD *
```

```
DMCL=dmc1-name
DBNAME=dbname or segment-name
```

```
//SYSIPT DD *
```

```
ASF-data-area-name ASF-data-area-name
```

---

<i>idms.dba.loadlib</i>	Data set name of the load library containing the DMCL and database name table load modules
-------------------------	--

---

<i>idms.custom.loadlib</i>	Data set name of the load library containing customized executable CA IDMS load modules
----------------------------	---

---

<i>idms.cagjload</i>	Data set name of the load library containing executable CA IDMS modules that do not require customization
<i>asfdict</i>	DDname of the ASF (DDL DML) dictionary area
<i>idms.asfdict.ddldml</i>	Data set name of the ASF (DDL DML) dictionary area
<i>dcmmsg</i>	DDname of the system message (DDLDCMSG) area
<i>idms.sysmsg.ddldcmmsg</i>	Data set name of the system message (DDLDCMSG) area
<i>sysjml</i>	DDname of a tape journal file, if journaling
<i>idms.tapejml</i>	Dataset name of tape journal file
<i>sys004</i>	DDname of the output file for the subschema produced by the IDMSRSSM program
<i>output.file</i>	Data set name of the output file for the subschema produced by the IDMSRSSM program
<i>dmcl-name</i>	The name of a DMCL
<i>dbname or segment-name</i>	Database name or segment name identifying the DDL DML area of the ASF dictionary
<i>ASF-data-area-name</i>	The name of an ASF extent area

To run IDMSRSSM under the central version, add a SYSCTL file.

**z/VSE JCL**

**IDMSRSSM**

```
// JOB IDMSRSSM
// LIBDEF *,SEARCH=idms libraries
// EXEC PROC=IDMSLBLS
// ASSGN sysnnn,DISK,VOL=nnnnnn,SHR
// DLBL SYS004,'output.file'
// EXEC IDMSRSSM,SIZE=1048K
DMCL=dmcl-name
DBNAME=dbname or segment-name
/*
AREA=extent-area-name AREA=extent-area-name
/*
```

*IDMSLBLS* Name of the procedure provided at installation that contains the file definitions for CA IDMS dictionaries, databases, disk journal files, and the SYSIDMS file.

For a complete listing of IDMSLBLS, see the *CA IDMS Conversion Guide*

<i>sys004</i>	Filename of the output file for the subschema produced by IDMSRSSM program
<i>dmcl-name</i>	The name of a DMCL
<i>dbname or segment-name</i>	Database name or segment name identifying the DDLML area of the ASF dictionary
<i>extent-area-name</i>	The name of an ASF extent area

To run IDMSRSSM under the central version, add a SYSCTL file.

### VM/ESA Commands

#### IDMSRSSM

```
FILEDEF SYSLST PRINTER
FI asfdict DISK cdms asfdict fm (RECFM F LRECL ppp BLKSIZE ppp XTENT nnn)
FI dcmsg DISK cdms dmsgdb fm (RECFM F LRECL ppp BLKSIZE ppp XTENT nnn)
FILEDEF sysjml TAP1 SL VOLID nnnnn (RECFM VB LRECL lll BLKSIZE bbb)
FILEDEF SYS004 DISK sys004 output a
FILEDEF SYSIDMS DISK sysidms input a
FILEDEF SYSIPT DISK idmsrsm input a
GLOBAL LOADLIB dbalib idmslib
OSRUN IDMSRSSM
```

<i>asfdict</i>	DDname of the ASF (DDLML) dictionary area
<i>cdms asfdict fm</i>	File identifier of the ASF (DDLML) dictionary area
<i>dcmsg</i>	DDname of the system message (DDLDCMSG) area
<i>cdms dmsgdb fm</i>	File identifier of the system message (DDLDCMSG) area
<i>ppp</i>	Page size of the database file
<i>nnn</i>	Number of pages in the database file
<i>sysjml</i>	DDname of the tape journal file, if journaling
<i>nnnnn</i>	Volume serial number of the tape journal file
<i>lll</i>	Record length of the tape journal file
<i>bbb</i>	Block size of the tape journal file
<i>sys004 output a</i>	File identifier of file containing subschema produced by IDMSRSSM program
<i>sysidms input a</i>	File identifier of the file containing the following: <ul style="list-style-type: none"> <li>■ <i>dmcl-name</i>— Name of the DMCL</li> <li>■ <i>segment-name</i>— database name or segment name identifying the DDLML area of the ASF dictionary</li> </ul>



---

<i>idmsrsm input a</i>	File identifier of the file containing the names of the ASF extent areas you will migrate
<i>dbalib</i>	Filename of the load library containing the DMCL and database name table load modules
<i>idmslib</i>	Filename of the load library containing executable CA IDMS modules

---

## Output from IDMSRUPD

### Output from IDMSRUPD

The IDMSRUPD program produces an output listing identifying the records processed and whether or not the RDEFREC was updated with a db-key.

### Sample IDMSRUPD Output Listing

A portion of the output listing produced by the IDMSRUPD program is provided below.

All errors that result from the execution of the IDMSRUPD program are due to a mismatch between the OOK record and the table definition. When any error results, the RDEF record is *not* updated with a db-key.

Date/time in OOK record is not the same as found in the RDEF record.

Status of table is not 'G' for generated.

```
STARTING TO UPDATE IDMSR-AREA
PROCESSING RECORD ID: 31998  RFUR-000102-00AK
DBKEY UPDATED IN RDEFREC FOR TDN: 000102

PROCESSING RECORD ID: 31996  RFUR-000101-00AK
TABLE NOT GENERATED FOR TDN: 000101

PROCESSING RECORD ID: 31994  RFUR-000103-00AK
DBKEY UPDATED IN RDEFREC FOR TDN: 000103

PROCESSING RECORD ID: 31992  RFUR-000104-00AK
DBKEY UPDATED IN RDEFREC FOR TDN: 000104

PROCESSING RECORD ID: 31990  RFUR-000106-00AK
DBKEY UPDATED IN RDEFREC FOR TDN: 000106

PROCESSING RECORD ID: 31988  RFUR-000105-00AK
RECORD: RFUR-000105-00AK
NOT FOUND IN AREA: IDMSR-AREA2

PROCESSING RECORD ID: 31986  RFUR-000108-00AK
DATE/TIME MISMATCH FOR TDN: 000108

PROCESSING RECORD ID: 31984  RFUR-000114-00AK
DATE/TIME MISMATCH FOR TDN: 000114

PROCESSING RECORD ID: 31982  RFUR-000109-00AK
DBKEY UPDATED IN RDEFREC FOR TDN: 000109

PROCESSING RECORD ID: 31980  RFUR-000111-00AK
DBKEY UPDATED IN RDEFREC FOR TDN: 000111

PROCESSING RECORD ID: 31974  RFUR-000137-00AK
DBKEY UPDATED IN RDEFREC FOR TDN: 000137

PROCESSING RECORD ID: 31972  RFUR-000115-00AK
DBKEY UPDATED IN RDEFREC FOR TDN: 000115

PROCESSING RECORD ID: 31970  RFUR-000116-00AK
DBKEY UPDATED IN RDEFREC FOR TDN: 000116

PROCESSING RECORD ID: 31968  RFUR-000117-00AK
TABLE NOT GENERATED FOR TDN: 000117

PROCESSING RECORD ID: 31966  RFUR-000118-00AK
DBKEY UPDATED IN RDEFREC FOR TDN: 000118

PROCESSING RECORD ID: 31964  RFUR-000205-00AK
DBKEY UPDATED IN RDEFREC FOR TDN: 000205

PROCESSING RECORD ID: 31962  RFUR-000120-00AK
DBKEY UPDATED IN RDEFREC FOR TDN: 000120

PROCESSING RECORD ID: 31960  RFUR-000121-00AK
DBKEY UPDATED IN RDEFREC FOR TDN: 000121
```

# Index

---

## \$

- \$OBJECT-SECURITY-NAME-TABLE\$ • 102, 173, 232
  - generating • 173
  - owner • 102
  - table name • 102
  - table-definition number • 102
- \$SECURITY-RUNTIME-TABLE\$ • 102, 105, 173, 232
  - generating • 173
  - Load Data option • 105
  - owner • 105
  - security name • 102, 105
  - selection criteria • 102, 105
  - user • 102, 105
- \$SRT-OST-CROSS-REFERENCE\$ • 102, 173, 232
  - generating • 173
- \$UNSTRUCTURED-IDB-OBJECT\$ • 232

## [

- [Clear] key • 38, 41
- [Enter] key • 38
- [PA1] key • 38, 188
- [PA2] key • 38
- [PF1] key • 38
- [PF7] key • 38
- [PF8] key • 38

## A

- accessing ASF • 40
- accessing tables • 91, 126
- activity selection • 188
- Activity Selection screen • 32, 51, 103, 188
- Add passkey • 217
- adding columns • 81, 155
- additional keys • 152, 251
- administrative authority • 36, 165, 188, 262
- Administrator Control screen • 165, 259
- allocating space • 263
- APPLICANT table • 69
- application • 17, 48, 57, 59, 74, 85, 126, 191, 241
  - generating • 57, 85, 191, 241
  - limiting • 126
  - outlining • 48
- area • 124, 166, 171, 241, 263
  - adding • 171

- assignment • 124
- definition • 166
- user default • 263
- ASEL • 32, 51, 188
- ASF • 30, 115, 166
  - initializing • 166
  - processing • 115
  - screens • 30
  - structure • 30
- ASF-defined tables • 21
- assigning passkeys • 92
- attributes • 136, 245

## B

- BROWSE passkey • 93, 218

## C

- CA ADS • 184
- CA Culprit • 183
- CA OLQ • 66
- calculation • 81, 130, 133, 245
  - examples • 133
  - expressions • 130
- catalog • 115, 179, 186, 262
  - deleting all object entities • 179, 262
- catalog access passkey • 91, 93, 218
- Catalog Control screen • 165, 262
- CDEF • 53, 77, 155, 196
- change access • 241
- changing data • 62, 89
- CNTL • 165, 259
- code table • 135, 245
- column • 19, 48, 50, 53, 59, 76, 77, 81, 89, 128, 130, 145, 150, 152, 191, 196, 245
  - adding • 81, 196
  - calculation • 130
  - defining • 48, 53, 191, 196
  - deleting • 77, 196
  - derived • 145
  - display limitations • 196
  - display sequence • 196
  - displayed name • 245
  - excluding • 77
  - extended definition • 196, 245
  - external picture • 245

---

- format • 128, 245
- index keys • 150
- initial value • 245
- internal picture • 245
- join • 50, 76, 89, 152, 191
- level • 130, 245
- naming • 196
- stored name • 245
- type • 196
- unique key • 196
- usage • 245
- values • 130
- width • 196
- work • 81
- column attributes • 245
- column definition • 53, 77, 155, 196
  - modifying • 53, 155, 196
- Column Definition screen • 53, 77, 155, 196
- column ID number • 200
- concatenated key • 152, 251
- conditional operator • 191, 200, 254
  - compound • 200
- Copy passkey • 217
- Create passkey • 217
- CTLG • 165, 262
- CULL DBA user ID • 168
- currency • 53, 196
- currency?Specify C (CURRENCY) when the column is to contain • 196
- cursor positioning • 39

## D

- data • 22, 34, 59, 62, 63, 89, 188, 209
  - access • 34
  - changing • 62, 89, 188, 209
  - current picture • 22
  - deleting • 63
  - displaying • 62, 188, 209
  - loading • 188, 209
  - selecting • 63
  - storing • 59
- data access passkeys • 91, 93, 217
- data defined tables • 141
- Data Display screen • 59, 62, 86, 209
- Data Selection screen • 63, 87, 200
- data table • 17, 19
- database • 17, 115, 259
  - administration • 259

- definition area • 115
  - private • 17
  - production • 17
- DBCS • 191, 196
- DDIS • 62, 86, 209
- default values • 53
- defaults • 40, 128, 153, 169, 259, 263
  - area • 263
  - model dialog • 259, 263
  - model subschema • 259, 263
  - models • 169, 259, 263
  - storage sequence • 153
  - system • 169, 259
  - user • 169, 263
- defining columns • 53
- defining selection criteria • 250
- defining tables • 51
- defining views • 73
- definition area • 115, 166
- definition number • 191
- Delete Passkeys screen • 99, 223
- deleting access • 241
- deleting data • 63
- deleting passkeys • 99
- deleting tables • 160, 191
- derivation • 76, 139, 191, 254
  - extended • 254
  - join criteria • 139
- dialog • 119, 169, 241, 259, 263
  - model • 169, 259, 263
- directory • 43, 91, 188
  - access to • 91
  - list • 188
  - with CA-ICMS • 43
- display access • 241
- display format • 128, 196, 245
- display sequence • 53, 77, 153, 196, 251
- Display/Change Row-Level Security screen • 238
- displayed column name • 245
- displaying data • 62, 86
- DLOD • 59, 209
- DML • 183
- DPAS • 99, 223
- DSEL • 63, 87, 200
- duplicates • 152

## E

- edit table • 135, 245

---

efficiency • 251  
element • 130, 245  
Erase passkey • 217  
error messages • 191  
    displaying • 191  
errors, row-level security • 107, 178  
Extended Column Definition screen • 127, 245  
extended functions • 123  
Extended Key Definition screen • 150, 155, 251  
extended screens • 241  
Extended Selection Definition screen • 83, 137, 155, 250  
Extended Table Definition screen • 123, 155, 241  
Extended Table Derivation screen • 139, 155, 254  
external picture • 128, 145, 245

## F

fast access keys • 152, 251  
field highlighting • 39  
fixed-format expressions • 137, 200  
folders • 43, 188  
    listing contents of • 188  
    name • 188  
freeform expressions • 137, 200  
functions • 30, 33, 34, 35, 36, 41, 43, 51, 85, 93, 123, 165  
    administrative • 36, 165  
    data access • 34  
    extended • 123  
    generate • 85  
    passkey • 35  
    selecting • 41, 43, 51, 93  
    table definition • 33

## G

Give Passkeys screen • 93, 222  
GPAS • 93, 222  
group • 107, 130, 245  
    assigning row-level security • 107  
group fields • 145

## H

HELP • 38, 186, 188, 191, 212, 220, 222, 223, 225, 227, 232, 233, 235, 251, 254  
    generic column name • 235  
Help screen • 33, 212  
highlighting • 39

## I

identifying sources • 139  
IDMSDBLP • 290, 313  
    IDMSRADM • 290  
    MAINTAIN ASF • 313  
IDMSLBLE • 295  
IDMSRADM • 288, 289, 290, 293  
    IDMSDBLP • 290  
    JCL (z/OS? CV mode) • 288  
    JCL (z/OS? Local mode) • 289  
    JCL (z/VM) • 293  
    JCL (z/VSE) • 290  
IDMSRADM utility • 271, 281  
    deleting post office objects • 281  
IDMSRSSM program • 343, 350  
    execution JCL • 343, 350  
IDMSRUPD program • 353  
    output • 353  
indexes • 150, 152, 155, 251  
    additional • 251  
    concatenated keys • 152, 251  
    duplicates option • 152  
    modifying • 155  
    single key • 251  
    sort order • 152, 251  
Information Center Management System • 100, 115, 182, 188  
    folders • 188  
initial value • 130, 245  
initializing • 166, 173, 259  
    ASF • 166  
    row-level security • 173  
integration • 115  
internal picture • 128, 245

## J

JCL • 295, 343, 350  
    IDMSLBLE • 295  
    IDMSRSSM program • 343, 350  
JCL (z/OS ? CV mode) • 288  
    IDMSRADM • 288  
JCL (z/OS? Local mode) • 289  
    IDMSRADM • 289  
JCL (z/VM) • 293, 315  
    IDMSRADM • 293  
    MAINTAIN ASF • 315  
JCL (z/VSE) • 290, 313  
    IDMSRADM • 290

---

MAINTAIN ASF • 313  
JCL ? Local mode) • 311  
MAINTAIN ASF • 311  
join • 23, 48, 50, 74, 76, 89, 139, 142, 152, 191, 254  
column • 48, 50, 76, 89, 152, 191, 254  
conditions • 191  
criteria • 139, 191  
set • 142, 254

## K

key • 251  
concatenated • 251  
fast access • 251  
index • 251  
multiple • 251  
single • 251  
sort order • 251

## L

List passkey • 217  
load access • 241  
Load Row-Level Security screen • 235  
loading data • 59  
locking users • 178, 262  
logical record • 241  
LRF • 319

## M

MAINTAIN ASF • 311, 313, 315  
IDMSDBLP • 313  
JCL (z/OS? Local mode) • 311  
JCL (z/VM) • 315  
JCL (z/VSE) • 313  
Management passkey • 218  
map • 119, 135, 169, 241, 259, 263  
characteristics • 135  
model • 169, 259, 263  
message field • 37, 39  
Message screen • 33, 211  
messages • 191  
models • 169  
dialog • 169  
map • 169  
subschema • 169  
Modify passkey • 217  
modifying column definitions • 53  
MSGs • 211

## N

numeric column?Specify N (NUMERIC) when the column is to contain • 196

## O

occurring fields • 145  
OnLine Query • 33, 66, 182  
OOAK record • 118  
operations, relational • 17, 23  
operator • 254

## P

PA key • 38  
predefined • 38  
paging • 188, 196, 251, 254  
PASG • 97, 225  
PASH • 98, 227  
PASS • 93  
passkey • 35, 43, 91, 92, 93, 95, 97, 99, 100, 172, 215, 216, 217, 218, 219, 222, 223, 225, 226, 227, 228, 263  
assigning • 92, 216  
authorization • 216, 263  
BROWSE • 93  
catalog access • 218, 219  
data access • 93, 217, 219  
deleting • 99, 223  
displaying • 97, 225, 226, 227, 228  
giving • 222  
implicit • 217  
limiting authorization • 172  
predefined combinations • 91, 95, 219  
screens • 215  
security • 216  
summary • 100  
Passkey Function screen • 93  
Passkey Function screen • 220  
Passkey Function screen  
i1.PASS The Passkey Function screen • 220  
Passkeys Given screen • 97, 226  
Passkeys Held screen • 98, 228  
password • 186  
PF-key • 37, 38  
global • 38  
local • 38  
predefined • 38  
simulating • 37, 38

---

post office object, deleting • 281  
predefined passkey combinations • 91, 95, 219  
private database • 17  
processing • 115  
production database • 17  
project • 23, 74, 77

## Q

QUERY • 66

## R

RDIS • 238  
record • 241  
    logical • 241  
record descriptions • 119  
Redefine passkey • 217  
redefining tables • 155  
regenerating tables • 158  
relational definition records • 115, 166  
relational operations • 17, 74  
RLOD • 235  
row • 19, 59, 62, 63, 83, 87, 89, 153, 196, 251  
    changing • 62, 89  
    deleting • 63  
    display sequence • 153, 196, 251  
    selecting • 63, 83, 87  
    sorted • 153  
    storage sequence • 153, 251  
row-level security • 37, 101, 102, 103, 105, 106, 107,  
109, 112, 155, 173, 176, 178, 231  
    authorization • 101, 231  
    considerations • 112  
    corporate objects • 101  
    errors, correcting • 107, 178  
    establishing • 173  
    generic column name • 109  
    groups, assigning • 107  
    groups, deleting • 109  
    groups, modifying • 109  
    implementing • 101  
    initializing • 101  
    limiting authorization • 173  
    modifying • 155  
    private objects • 101  
    screens • 231  
    security name • 101, 103  
    selection criteria • 101, 105  
    system tables • 102, 173

    tables, assigning to • 102  
    users, associating • 105  
    users, authorizing • 176  
    validation • 106  
Row-Level Security screen • 103, 155, 233  
row-level security selection • 107  
    status • 107  
Row-Level Security Selection screen • 236  
RSEC • 155, 233  
RSEL • 236

## S

schema • 166, 254  
    system-supplied • 166  
    version number • 254  
screen • 37, 38, 188, 212  
    format • 37  
    Help • 212  
    management • 38  
    return to prior • 188  
screen name • 41  
screen name field • 37  
screen size • 196  
screens • 30, 33, 41, 185, 241  
    basic • 185  
    definition • 33  
    extended • 241  
    moving between • 41  
secondary area • 171  
    adding • 171  
security • 216, 231  
security name • 108, 109  
    considerations • 109  
    delete • 109  
    groups • 108  
    modify • 109  
see=select,project,join relational operations • 23  
see=storedtable..stored • 47  
    see=storedtable.stored • 47  
select • 23, 74  
selecting data • 188  
selecting rows • 63, 83, 87  
selecting tables • 188  
selection criteria • 101, 106, 137, 200  
    fixed-format • 137, 200  
    freeform • 137, 200  
    row-level security • 106  
selection definition • 155

- modifying • 155
- session • 40, 43
  - beginning • 40
  - ending • 43
- set descriptions • 119
- Show Passkeys Given screen • 97, 225
- Show Passkeys Held screen • 98, 227
- SIGN • 32, 186
- signing off • 43
- signing on • 40
- Signon screen • 32, 40, 186
- simulated PF-key field • 37, 38
- sort order • 153, 251
- source table • 86, 139, 141
- source table?Enter information about the • 191
  - access path?Enter the data table name (or access path) • 191
- source tables • 161
  - deleting • 161
  - modifying • 161
- space allocation • 263
  - user defaults • 263
- status • 53, 57, 191
- STATUS column • 107
- status, row-level security • 155
- storage format • 128, 196
- storage sequence • 153, 196, 251
- stored column name • 245
- stored format • 245
- stored table • 19, 21, 47, 51, 69
  - defining • 51
  - review • 69
- subschema • 119, 169, 254, 259, 263
  - model • 169, 259, 263
- SYST • 165, 232, 268
- system defaults • 169, 259
- system messages • 39
- System Table Maintenance screen • 37, 165, 232, 268
- system tables • 102, 173
  - §OBJECT-SECURITY-NAME-TABLE§ • 102
  - §SECURITY-RUNTIME-TABLE§ • 102
  - §SRT-OST-CROSS-REFERENCE§ • 102
  - generating • 173
  - OST • 102
  - SRT • 102
  - SXRF • 102
- system-supplied schema • 115

## T

- table • 19, 21, 22, 43, 48, 51, 53, 57, 59, 73, 74, 85, 86, 90, 91, 102, 123, 126, 139, 150, 155, 158, 160, 188, 191, 196, 200, 218, 250, 251, 254
  - access • 188, 218
  - columns • 48, 53, 59
  - complex derivation • 254
  - defining • 51, 102, 188, 191
  - deleting • 160, 191
  - derivation, simple • 191
  - deriving • 73, 139, 191, 254
  - description • 191
  - directory • 188
  - display format • 196
  - display sequence • 251
  - existing • 188
  - format • 196
  - generating • 57, 85, 158, 191
  - granting access • 91
  - indexes • 150, 251
  - limiting use of • 126
  - managing • 155
  - naming • 43, 51, 74
  - new • 188, 191
  - outlining • 48
  - owner • 188
  - redefining • 196
  - regenerating • 102
  - review • 90
  - row-level security • 102
  - rows • 59
  - selection • 43, 200, 250
  - sorting • 251
  - source • 191
  - status • 53, 57, 191
  - storage format • 196
  - storage sequence • 251
  - stored • 19, 21, 191
  - type • 191
  - underlying characteristics • 123
  - ungenerated • 191
  - using • 59, 86
  - view • 19, 22, 191
- table definition • 33, 115, 155, 172, 191, 241, 259
  - creation time • 241
  - deleting • 191
  - extended • 241
  - length • 241



---

- modifying • 155
- reviewing • 191
- unlocking • 172, 259
- update time • 241
- updating • 191
- table definition number • 115, 181, 191
- Table Definition screen • 51, 155, 191
- table derivation • 155
  - modifying • 155
- table directory • 43
  - with CA-ICMS • 43
- table selection • 188
- table storage • 118
- TDEF • 51, 155, 191
- text column?Specify T (TEXT) when the column is to contain alphabetic, • 196
- total space allocation • 263
- type • 196
- type, column • 53, 196
  - changing • 196
  - currency • 53

## U

- ULIM • 165, 263
- unique key • 48, 50, 53, 150, 196
  - stored table • 50
  - view • 50
- USER • 165, 267
- user defaults • 169
- user ID • 186
- User Limits and Defaults screen • 165, 263
- User Maintenance screen • 165, 267
- users • 178
  - locking and unlocking • 178
- using tables • 59, 86

## V

- view • 19, 22, 50, 73, 85, 90, 141
  - defining • 73, 141
  - generating • 85
  - join column • 50
  - retrieval • 22
  - review • 90
  - update • 22
  - use of • 22

## W

- work column • 130, 245

## X

- XCOL • 127, 245
- XDER • 139, 155, 254
- XKEY • 150, 155, 251
- XSEL • 83, 137, 155, 250
- XTAB • 123, 155, 241