

Cincom

SUPRA SERVER PDM

Directory Views
(VMS)

P25-1120-43



SUPRA[®] Server PDM Directory Views (VMS)

Publication Number P25-1120-43

© 1983 – 1990, 1993, 1998–2002 Cincom Systems, Inc.
All rights reserved

This document contains unpublished, confidential, and proprietary information of Cincom. No disclosure or use of any portion of the contents of these materials may be made without the express written consent of Cincom.

The following are trademarks, registered trademarks, or service marks of Cincom Systems, Inc.:

AD/Advantage [®]	iD CinDoc [™]	MANTIS [®]
C+A-RE [™]	iD CinDoc Web [™]	Socrates [®]
CINCOM [®]	iD Consulting [™]	Socrates [®] XML
Cincom Encompass [®]	iD Correspondence [™]	SPECTRA [™]
Cincom Smalltalk [™]	iD Correspondence Express [™]	SUPRA [®]
Cincom SupportWeb [®]	iD Environment [™]	SUPRA [®] Server
CINCOM SYSTEMS [®]	iD Solutions [™]	Visual Smalltalk [®]
	intelligent Document Solutions [™]	VisualWorks [®]
gOOj [™]	Intermax [™]	

UniSQL[™] is a trademark of UniSQL, Inc.
ObjectStudio[®] is a registered trademark of CinMark Systems, Inc.

All other trademarks are trademarks or registered trademarks of their respective companies.

Cincom Systems, Inc.
55 Merchant Street
Cincinnati, Ohio 45246-3732
U.S.A.

PHONE: (513) 612-2300
FAX: (513) 612-2000
WORLD WIDE WEB: <http://www.cincom.com>

Attention:

Some Cincom products, programs, or services referred to in this publication may not be available in all countries in which Cincom does business. Additionally, some Cincom products, programs, or services may not be available for all operating systems or all product releases. Contact your Cincom representative to be certain the items are available to you.

Release information for this manual

The *SUPRA Server PDM Directory Views (VMS)*, P25-1120-43, is dated January 15, 2002. This document supports Release 2.2 of SUPRA Server for VMS environments.

We welcome your comments

We encourage critiques concerning the technical content and organization of this manual. Please take the [survey](#) provided with the online documentation at your convenience.

Cincom Technical Support for SUPRA Server PDM

FAX: (513) 612-2000
Attn: SUPRA Server Support

E-mail: helpna@cincom.com

Phone: 1-800-727-3525

Mail: Cincom Systems, Inc.
Attn: SUPRA Server Support
55 Merchant Street
Cincinnati, OH 45246-3732
U.S.A.



Contents

About this book	vii
Using this document.....	vii
Document organization	vii
Conventions	viii
SUPRA Server documentation series	xi
Introduction to Directory Views	13
Setting up the global view file and logical names.....	14
Descriptions of the Directory Views.....	15
Accessing Directory Views from SPECTRA, DBAID and MANTIS.....	23
Accessing Directory Views from SPECTRA.....	24
Accessing Directory Views from DBAID.....	25
Accessing Directory Views from MANTIS.....	26
Using Directory Views effectively	27
Views accessing entity details	29
UE-BUFFER.....	30
UE-DATA-ITEM.....	31
UE-DATA-SET	34
UE-DATABASE-DESCRIPTION	36
UE-DOMAIN.....	39
UE-FILE-SPEC.....	43
UE-INDEX	45
UE-INDEX-FILE-SPEC	46
UE-LOGICAL-DATA-ITEM.....	48
UE-LOGICAL-VIEW	49
UE-PROGRAM	51
UE-RECORD	53
UE-RMS-DATA-SET	55
UE-RMS-FILE-SPEC	57
UE-RMS-KEY	59
UE-SECONDARY-KEY	61
UE-SYSTEM-LOG	63
UE-TASK-LOG.....	64
UE-USER	65
UE-VALIDATION-TABLE	66

Views accessing the relationships between entities	67
UX-BUFFER-OF-DATA-SET	68
UX-BUFFER-OF-DBDESC	69
UX-DATA-ITEM-IN-SEC-KEY	70
UX-DATA-ITEM-OF-RECORD	71
UX-DATA-SET-OF-DBDESC	73
UX-DERIVED-VIEW-CONSTANT	74
UX-DOMAIN-OF-DATA-ITEM	75
UX-DOMAIN-USES-VALIDATION-TABL	76
UX-FILE-SPEC-OF-DATA-SET	77
UX-FILE-SPEC-OF-DATABASE	78
UX-FILE-SPEC-OF-INDEX	79
UX-INDEX-IN-DATA-SET	80
UX-INDEX-OF-DBDESC	81
UX-LINKPATH-OF-RECORD	82
UX-LOGICAL-DATA-ITEM-OF-DATA-S	84
UX-LOGICAL-DATA-ITEM-OF-LV	85
UX-LOGICAL-VIEW-OF-DBDESC	87
UX-PHYSICAL-TO-LOGICAL-DATA-IT	88
UX-PROGRAM-USES-LOGICAL-VIEW	89
UX-RECORD-OF-DATA-SET	90
UX-RMS-KEY-IN-DATA-ITEM	91
UX-RMS-KEY-IN-DATA-SET	92
UX-SEC-KEY-IN-INDEX	93
UX-SUB-DATA-ITEM	94
UX-SYSTEM-LOG-OF-DBDESC	95
UX-TASK-LOG-OF-DBDESC	96
UX-USER-MAY-RUN-PROGRAM	97
UX-USER-MAY-USE-LOGICAL-VIEW	98
UX-VIEW-TO-VIEW	99
Special views	101
Views that access comments	102
A view that provides access definitions	104
Views that show alternative relationship entities	104
UR-PHYSICAL-LOGICAL-DATA-ITEM	105
UR-VIEW-TO-VIEW	106
A view that uses two keys to access constant values	108
Data type descriptions	109
Index	123

About this book

Using this document

This document is written for database administration personnel who are familiar with SUPRA database design and maintenance.

SUPRA Server PDM Directory Views (VMS), P25-1120, describes a set of Cincom-supplied views of the Directory database. These views describe the entities held on the Directory, providing read-only access to all user database definitions. Use the Directory Views to report on the entities and their relationships defined on your SUPRA Directory.

Document organization

The information in this manual is organized as follows:

Chapter 1—Introduction to Directory Views

Describes the global views file and how to access Directory Views.

Chapter 2—Views accessing entity details

Describes the twenty entity views, which access details of the entities defined on the directory.

Chapter 3—Views accessing the relationships between entities

Describes Directory Views that provide information about relationships between entities.

Chapter 4—Special views

Describes views that fall into neither the Entity category nor the Relationship category.

Appendix—Data type descriptions

Provides data type descriptions of the view columns for the COBOL, FORTRAN, and BASIC programming languages.

Index

Conventions

The following table describes the conventions used in this document series:

Convention	Description	Example
Constant width type	Represents screen images and segments of code.	<pre>PUT 'customer.dat' GET 'miller\customer.dat' PUT '\DEV\RMT0'</pre>
Slashed b (<i>b</i>)	<p>Indicates a space (blank).</p> <p>The example indicates that four spaces appear between the keywords.</p>	<pre>BEGNbbbSERIAL</pre>
Brackets []	<p>Indicate optional selection of parameters. (Do not attempt to enter brackets or to stack parameters.) Brackets indicate one of the following situations:</p>	
	<p>A single item enclosed by brackets indicates that the item is optional and can be omitted.</p> <p>The example indicates that you can optionally enter a WHERE clause.</p>	<pre>[WHERE <i>search-condition</i>]</pre>
	<p>Stacked items enclosed by brackets represent optional alternatives, one of which can be selected.</p> <p>The example indicates that you can optionally enter either WAIT or NOWAIT. (WAIT is underlined to signify that it is the default.)</p>	<pre>[<u>(WAIT)</u> (NOWAIT)]</pre>

Convention	Description	Example
Braces { }	<p>Indicate selection of parameters. (Do not attempt to enter braces or to stack parameters.) Braces surrounding stacked items represent alternatives, one of which you must select.</p> <p>The example indicates that you must enter ON or OFF when using the MONITOR statement.</p>	<pre>MONITOR { ON } { OFF }</pre>
<u>Underlining</u> (In syntax)	<p>Indicates the default value supplied when you omit a parameter.</p> <p>The example indicates that if you do not choose a parameter, the system defaults to WAIT.</p>	<pre>[(WAIT)] [(NOWAIT)]</pre>
	<p>Underlining also indicates an allowable abbreviation or the shortest truncation allowed.</p> <p>The example indicates that you can enter either STAT or STATISTICS.</p>	<pre><u>STATISTICS</u></pre>
Ellipsis points...	<p>Indicate that the preceding item can be repeated.</p> <p>The example indicates that you can enter multiple host variables and associated indicator variables.</p>	<pre>INTO :host-variable [:ind- variable],...</pre>

Convention	Description	Example
UPPERCASE lowercase	In most operating environments, keywords are not case-sensitive and they are represented in uppercase. You can enter them in either uppercase or lowercase.	COPY MY_DATA.SEQ HOLD_DATA.SEQ
	In the UNIX operating environment, keywords are case-sensitive and you must enter them exactly as shown.	cp *.QAR /backup
<i>Italics</i>	Indicate variables you replace with a value, a column name, a file name, and so on. The example indicates that you must substitute the name of a table.	FROM <i>table-name</i>
Punctuation marks	Indicate required syntax that you must code exactly as presented. () parentheses . period , comma : colon ' ' single quotation marks	<i>(user-id, password, db-name)</i> INFILE 'Cust.Memo' CONTROL LEN4
SMALL CAPS	Represent a required keystroke. Multiple keystrokes are hyphenated.	ALT-TAB

SUPRA Server documentation series

SUPRA Server is the advanced relational database management system for high-volume, update-oriented production processing. A number of tools are available with SUPRA Server including DBA Functions, DBAID, precompilers, SPECTRA, and MANTIS. The following list shows the manuals and tools used to fulfill the data management and retrieval requirements for various tasks. Some of these tools are optional. Therefore, you may not have all the manuals listed. For a brief synopsis of each manual, refer to the *SUPRA Server PDM Digest for VMS Systems*, P25-9062.

Overview

- ◆ *SUPRA Server PDM Digest for VMS Systems*, P25-9062

Getting started

- ◆ *SUPRA Server PDM VMS Installation Guide*, P25-0147
- ◆ *SUPRA Server PDM VMS Tutorial*, T25-2263

General use

- ◆ *SUPRA Server PDM Glossary*, P26-0675
- ◆ *SUPRA Server PDM Messages and Codes Reference Manual (PDM/RDM Support for UNIX & VMS)*, P25-0022

Database administration tasks

- ◆ *SUPRA Server PDM Database Administration Guide (UNIX & VMS)*, P25-2260
- ◆ *SUPRA Server PDM System Administration Guide (VMS)*, P25-0130
- ◆ *SUPRA Server PDM Utilities Reference Manual (UNIX & VMS)*, P25-6220
- ◆ *SUPRA Server PDM Directory Views (VMS)*, P25-1120
- ◆ *SUPRA Server PDM Windows Client Support User's Guide*, P26-7500*
- ◆ *SPECTRA Administrator's Guide*, P26-9220**

Application programming tasks

- ◆ *SUPRA Server PDM Programming Guide (UNIX & VMS)*, P25-0240
- ◆ *SUPRA Server PDM System Administration Guide (VMS)*, P25-0130
- ◆ *SUPRA Server PDM RDM Administration Guide (VMS)*, P25-8220
- ◆ *SUPRA Server PDM Windows Client Support User's Guide*, P26-7500*
- ◆ *MANTIS Planning Guide*, P25-1315**

Report tasks

- ◆ *SPECTRA User's Guide*, P26-9561**



Manuals marked with an asterisk (*) are listed twice because you use them for different tasks.



Educational material is available from your regional Cincom education department.

1

Introduction to Directory Views

The Directory Views are a set of predefined views of the SUPRA Directory database (SUPRAD). They provide read-only access to the entities and relationships held on the SUPRA Directory so that you can report on user-defined databases and views. Entities include user databases, PDM and RMS data sets, logical and physical data items, index and secondary key information, validation criteria, base and derived views, user names, programs, buffers, and recovery logs. Relationships are the connections between these entities.

For more information on your metadata that is displayed using the Directory Views, refer to the *SUPRA Server PDM Database Administration Guide (UNIX & VMS)*, P25-2260.

This chapter describes

- ◆ Setting up the global views file and logical names needed to access the Directory Views
- ◆ Accessing Directory Views from DBAID, SPECTRA and MANTIS
- ◆ Using Directory Views effectively by referring to the diagram of the Directory structure to identify which views to use

Setting up the global view file and logical names

To use the Directory Views, you need the following three files, which are located in the directory pointed to by the standard SUPRA logical SUPRA_REPORT:

- ◆ **DIRVIEWS.GBL.** Global view file containing the Directory Views. You must have access to this file.
- ◆ **DIRVIEWS.COM.** Example command file showing how to access the Directory Views from DBAID.
- ◆ **DIRVIEWS.EXP.** an exported SPECTRA process, BP1, showing how to access the Directory Views from a SPECTRA process.

Create the following logical definitions before you access the Directory Views:

- ◆ GVSHEMA pointing to the global view file, DIRVIEWS.GBL, that contains the views. Use standard SUPRA logicals, for example:

```
$ DEFINE GVSHEMA SUPRA_REPORT:DIRVIEWS.GBL
```

- ◆ CSI_SCHEMA identifying the SUPRAD database as the database to be accessed, for example:

```
$ DEFINE CSI_SCHEMA SUPRAD
```

where SUPRAD is a logical definition pointing to your directory database. The standard SUPRA logical definition using the PDM_LOGICALS_#####*.COM for your PDM system makes this definition. The variable ##### is either your 6-character UIC group number, which is all zeros if the PDM is systemwide, or the 1–8 character name of your multiple systemwide PDM. Refer to the *SUPRA Server PDM System Administration Guide (VMS)*, P25-0130, for more information.

- ◆ CSI_NODIRECTORY TRUE allowing applications to access the Directory Views held in the global view file without signing on to a SUPRA Directory, for example:

```
$ DEFINE CSI_NODIRECTORY TRUE
```

You need this definition because you cannot access the SUPRAD database if you are signed on to the SUPRA Directory.

Refer to the *SUPRA Server PDM System Administration Guide (VMS)*, P25-0130, for logical definition requirements when using prefixing.



If you omit the definition `CSI_NODIRECTORY TRUE`, any application attempting to sign on to the RDM looks for the database SUPRAD (identified by the logical name `CSI_SCHEMA`) on the SUPRA Directory. The attempted sign-on will fail with the message `USER DATABASE NOT IN DIRECTORY`.

Descriptions of the Directory Views

The sixty-two Directory Views fall into these categories:

- ◆ Views displaying entity details, described in “[Views accessing entity details](#)” on page 29. These views are identified by the prefix UE.
- ◆ Views displaying the relationships between entities, described in “[Views accessing the relationships between entities](#)” on page 67. These views are identified by the prefix UX.
- ◆ Special views providing comments (prefix UC), access definitions (prefix UN), relationships between entities using a different linkpath than the corresponding UX views (prefix UR), and a view that uses two keys to provide fast access (prefix US). Each of these are described in “[Special views](#)” on page 101.

Each chapter describes the views in the following format:

- ◆ Sample DBAID output to illustrate the records obtained by opening the view and typing either `GO*` or `GET*`.
- ◆ A brief description of the key to the view.
- ◆ Translation of any codes used to represent column values.
- ◆ Data type descriptions of each column in the view for use in designing MANTIS programs and SPECTRA processes that access the Directory Views, as described in “[Data type descriptions](#)” on page 109.

The following table lists the Directory Views in alphabetical order, gives a brief description of the information provided by each view, and shows you the section of this manual that displays the view:

Directory View name	Information retrieved
UC-BUFFER	Comments associated with each buffer. See “Data type descriptions” on page 109.
UC-DATA-ITEM	Comments associated with each physical data item. See “Data type descriptions” on page 109.
UC-DATA-SET	Comments associated with each data set. See “Data type descriptions” on page 109.
UC-DATABASE-DESCRIPTION	Comments associated with each database. See “Views that access comments” on page 102.
UC-DOMAIN	Comments associated with each domain. See “Data type descriptions” on page 109.
UC-FILE-SPEC	Comments associated with each physical file specification for a data set. See “Data type descriptions” on page 109.
UC-INDEX	Comments associated with each index. See “Data type descriptions” on page 109.
UC-INDEX-FILE-SPEC	Comments associated with each physical file specification. See “Data type descriptions” on page 109.
UC-LOGICAL-DATA-ITEM	Comments associated with each logical data item. See “Data type descriptions” on page 109.
UC-LOGICAL-VIEW	Comments associated with each view. See “Data type descriptions” on page 109.
UC-PROGRAM	Comments held on the Directory for each program. See “Data type descriptions” on page 109.
UC-RMS-DATA-SET	Comments associated with each RMS data set. See “Data type descriptions” on page 109.
UC-RMS-FILE-SPEC	Comments associated with each physical file specification for RMS data sets. See “Data type descriptions” on page 109.

Directory View name	Information retrieved
UC-RMS-KEY	Comments associated with each RMS key. See “Data type descriptions” on page 109.
UC-SECONDARY-KEY	Comments associated with each secondary key. See “Data type descriptions” on page 109.
UC-SYSTEM-LOG	Comments associated with each system log. See “Data type descriptions” on page 109.
UC-TASK-LOG	Comments associated with each task log. See “Data type descriptions” on page 109.
UC-USER	Comments associated with each username. See “Data type descriptions” on page 109.
UC-VALIDATION-TABLE	Comments associated with each validation table. See “Data type descriptions” on page 109.
UE-BUFFER	Buffer details including number of copies and type and size of the buffer. See “UE-BUFFER” on page 30.
UE-DATA-ITEM	Data item details including length, use, type, sign, displacement, subdata items, etc. See “UE-DATA-ITEM” on page 31.
UE-DATA-SET	Data set details including length, type, whether they are coded or uncoded, data set status, and primary key length. See “UE-DATA-SET” on page 34.
UE-DATABASE-DESCRIPTION	Database details as defined through the Database Details screen in DBA. See “UE-DATABASE-DESCRIPTION” on page 36.
UE-DOMAIN	Domain details as defined through the Domain Details screen in DBA. See “UE-DOMAIN” on page 39.
UE-FILE-SPEC	File specifications for PDM data sets including data set name, database name and physical file specification. See “UE-FILE-SPEC” on page 43.
UE-INDEX	Index name as defined in the Directory. See “UE-INDEX” on page 45.

Directory View name	Information retrieved
UE-INDEX-FILE-SPEC	Physical index file attributes including action taken if index is corrupt, location of null values in collating sequence, whether the PDM checks if the index is corrupt, and main and shadow file specifications. See “ UE-INDEX-FILE-SPEC ” on page 46.
UE-LOGICAL-DATA-ITEM	All logical data items defined on the Directory. See “ UE-LOGICAL-DATA-ITEM ” on page 48.
UE-LOGICAL-VIEW	Details of both base and derived views including view status, whether bound and, if bound, the bound view size, and the date and time the view was last saved. See “ UE-LOGICAL-VIEW ” on page 49.
UE-PROGRAM	Program details held on the Directory including program status, date and time it was enrolled on the Directory, the language it is written in and the modification level. See “ UE-PROGRAM ” on page 51.
UE-RECORD	Record details including type, record codes, number of data items, total length, etc. See “ UE-RECORD ” on page 53.
UE-RMS-DATA-SET	RMS data set details including length, type and status. See “ UE-RMS-DATA-SET ” on page 55.
UE-RMS-FILE-SPEC	RMS file specifications including data set name, database name, physical file specification and bucket size. See “ UE-RMS-FILE-SPEC ” on page 57.
UE-RMS-KEY	RMS key details including the key number, length, and uniqueness. See “ UE-RMS-KEY ” on page 59.
UE-SECONDARY-KEY	Secondary key details including the status of whether duplicate key values are permitted, the use of pointers to retrieve duplicate values, the sort direction and the expected percentage of duplicate values expected. See “ UE-SECONDARY-KEY ” on page 61.

Directory View name	Information retrieved
UE-SYSTEM-LOG	System log details including block size, number of blocks, and main and shadow file specifications for both system log components. See “ UE-SYSTEM-LOG ” on page 63.
UE-TASK-LOG	Task log details including block size, number of blocks, number of buffers, and main and shadow file specifications. See “ UE-TASK-LOG ” on page 64.
UE-USER	DBA user name details including access authority and encoded password. See “ UE-USER ” on page 65.
UE-VALIDATION-TABLE	Validation table name and status. See “ UE-VALIDATION-TABLE ” on page 66.
UN-LOGICAL-VIEW	Access definition for each logical view. See “ A view that provides access definitions ” on page 104.
UR-PHYSICAL-LOGICAL-DATA-ITEM	Logical to physical data item mapping using the logical data item name as the key. See “ UR-PHYSICAL-LOGICAL-DATA-ITEM ” on page 105.
UR-VIEW-TO-VIEW	Details of derived view using a base view. Uses the base view name as the key. See “ UR-VIEW-TO-VIEW ” on page 106.
US-DERIVED-VIEW-CONSTANT	Same as UX-DERIVED-VIEW-CONSTANT , but using two keys instead of one to provide quicker access. The keys are the derived view name and the column with the constant value. See “ A view that uses two keys to access constant values ” on page 108.
UX-BUFFER-OF-DATA-SET	Buffer details view that uses the data set name as the key. See “ UX-BUFFER-OF-DATA-SET ” on page 68.
UX-BUFFER-OF-DBDESC	Buffer details view that uses the database name as the key. See “ UX-BUFFER-OF-DBDESC ” on page 69.

Directory View name	Information retrieved
UX-DATA-ITEM-IN-SEC-KEY	Details of data items in a secondary key. Uses the secondary key name as the key. See “UX-DATA-ITEM-IN-SEC-KEY” on page 70.
UX-DATA-ITEM-OF-RECORD	Details of the data items in a record (data set). Uses the data set name and record type as the key. See “UX-DATA-ITEM-OF-RECORD” on page 71.
UX-DATA-SET-OF-DBDESC	Accesses the data sets in a database description. Uses the database name as the key. See “UX-DATA-SET-OF-DBDESC” on page 73.
UX-DERIVED-VIEW-CONSTANT	Same as US-DERIVED-VIEW-CONSTANT but uses only the derived view as the key. See “UX-DERIVED-VIEW-CONSTANT” on page 74.
UX-DOMAIN-OF-DATA-ITEM	Accesses domain details. Uses data item name as the key. See “UX-DOMAIN-OF-DATA-ITEM” on page 75.
UX-DOMAIN-USES-VALIDATION-TABL	Validation table details view that uses domain name as the key. See “UX-DOMAIN-USES-VALIDATION-TABL” on page 76.
UX-FILE-SPEC-OF-DATA-SET	File specification details view that uses data set name as the key. See “UX-FILE-SPEC-OF-DATA-SET” on page 77.
UX-FILE-SPEC-OF-DATABASE	File specification details view that uses database name as the key. See “UX-FILE-SPEC-OF-DATABASE” on page 78.
UX-FILE-SPEC-OF-INDEX	File specification details view that uses the index name as the key. See “UX-FILE-SPEC-OF-INDEX” on page 79.
UX-INDEX-IN-DATA-SET	Details of indices defined for a data set. Uses the data set name as the key. See “UX-INDEX-IN-DATA-SET” on page 80.
UX-INDEX-OF-DBDESC	Index details view that uses database name as the key. See “UX-INDEX-OF-DBDESC” on page 81.

Directory View name	Information retrieved
UX-LINKPATH-OF-RECORD	Details of the linkpaths in a record (data set). Uses the data set name and record type as the key. See “UX-LINKPATH-OF-RECORD” on page 82.
UX-LOGICAL-DATA-ITEM-OF-DATA-S	The logical data items defined for a data set. Uses the data set name as the key. See “UX-LOGICAL-DATA-ITEM-OF-DATA-S” on page 84.
UX-LOGICAL-DATA-ITEM-OF-LV	Details of logical data items used by a view. Uses view name as the key. See “UX-LOGICAL-DATA-ITEM-OF-LV” on page 85.
UX-LOGICAL-VIEW-OF-DBDESC	Views connected to a database. Uses database name as the key. See “UX-LOGICAL-VIEW-OF-DBDESC” on page 87.
UX-PHYSICAL-TO-LOGICAL-DATA-IT	The logical data item names for a physical data item. Uses physical data item name as the key. See “UX-PHYSICAL-TO-LOGICAL-DATA-IT” on page 88.
UX-PROGRAM-USES-LOGICAL-VIEW	The view used by a program. Uses program name as the key. See “UX-PROGRAM-USES-LOGICAL-VIEW” on page 89.
UX-RECORD-OF-DATA-SET	Details of the records in a data set. Uses data set name as the key. See “UX-RECORD-OF-DATA-SET” on page 90.
UX-RMS-KEY-IN-DATA-ITEM	Details of the RMS keys in a data item. Uses data item name as the key. See “UX-RMS-KEY-IN-DATA-ITEM” on page 91.
UX-RMS-KEY-IN-DATA-SET	Details of the RMS keys in a data set. Uses data set name as the key. See “UX-RMS-KEY-IN-DATA-SET” on page 92.

Directory View name	Information retrieved
UX-SEC-KEY-IN-INDEX	Details of secondary keys defined for an index. Uses index name as the key. See “UX-SEC-KEY-IN-INDEX” on page 93.
UX-SUB-DATA-ITEM	Subdata items using the subdefined data item name as the key. See “UX-SUB-DATA-ITEM” on page 94.
UX-SYSTEM-LOG-OF-DBDESC	System log details for a database. Uses database name as the key. See “UX-SYSTEM-LOG-OF-DBDESC” on page 95.
UX-TASK-LOG-OF-DBDESC	Task log details for a database. Uses database name as the key. See “UX-TASK-LOG-OF-DBDESC” on page 96.
UX-USER-MAY-RUN-PROGRAM	The programs available to a user. Uses user name as the key. See “UX-USER-MAY-RUN-PROGRAM” on page 97.
UX-USER-MAY-USE-LOGICAL-VIEW	The views available to a user. Uses user name as the key. See “UX-USER-MAY-USE-LOGICAL-VIEW” on page 98.
UX-VIEW-TO-VIEW	Details of the base views accessed by a derived view. Uses the derived view name as the key. See “UX-VIEW-TO-VIEW” on page 99.

Accessing Directory Views from SPECTRA, DBAID and MANTIS

Once you have set up the logical names described in “[Setting up the global view file and logical names](#)” on page 14, you can access the Directory Views with the user name DATABASE-DESCRIPTIONS and no password. Access the Directory Views from the following:

- ◆ SPECTRA
- ◆ DBAID
- ◆ MANTIS version 2.1 and above

Note that preprocessed RDML programs (BASIC, COBOL, FORTRAN) do not support access to the Directory Views.

Accessing Directory Views from SPECTRA

From SPECTRA, you can design processes to generate reports on selected entities and relationships. Sign on to SPECTRA with the user name DATABASE-DESCRIPTIONS and no password, and write SPECTRA processes to access your chosen views.

To illustrate the power of using SPECTRA processes to access Directory Views, Cincom provides the exported SPECTRA file DIRVIEWS.EXPEXP containing the sample SPECTRA process BP1. Import and execute the process BP1 as follows:

- ◆ Sign on to SPECTRA with the user name DATABASE-DESCRIPTIONS and no password.
- ◆ Type `IMPORT FROM DIRVIEWS.` at the SPECTRA command line (`==>`). Make sure the file DIRVIEWS.EXP is in the VMS Directory from which you invoke SPECTRA, or include the full path (`SUPRA_REPORT: DIRVIEWS.EXP`).
- ◆ Type `LP` (List processes) at the SPECTRA command line to check that the process BP1 has been imported.
- ◆ Type `SHOW BP1` at the SPECTRA command line to examine the contents of BP1.
- ◆ Type `GO BP1` at the SPECTRA command line to execute BP1.

After prompting you to enter the name of the database in uppercase and press RETURN, BP1 displays details of the physical and logical data items in the specified database. Refer to the *SPECTRA User's Guide*, P25-4560, for details of how to design and execute SPECTRA processes.

Accessing Directory Views from DBAID

From DBAID, you can open each Directory View in turn and type GO* or GET* to list the Directory information that is displayed with that view. The file DIRVIEWS.COM, which is in the directory pointed to by the logical SUPRA_REPORT, defines the necessary logical names, invokes DBAID, and opens a Directory View. Execute this file from DCL Command Level by typing

```
@SUPRA_REPORT:DIRVIEWS.COM
```

to see how to use a Directory View from DBAID. Alternatively, sign on to DBAID with the user name DATABASE-DESCRIPTIONS and no password, then open each view in turn. The following is an example DBAID session which opens the Directory View UX-DOMAIN-USES-VALIDATION-TABL:

```
$RUN CSVDBAID

                SUPRA RELEASE 1.0
                WELCOME TO DBAID - LEVEL 05
PLEASE SIGN ON
>DATABASE-DESCRIPTIONS
>Password:
FSI: * VSI: = MSG: SUCCESSFUL COMPLETION - SERVICE LEVEL 05
>OPEN UX-DOMAIN-USES-VALIDATION-TABL
FSI: * VSI: = MSG:                2544 BYTES USED IN OPENING VIEW.
>GO*

!                DOMN                !                VALIDATION-TABLE                !
!-----!-----!-----!-----!
!TEST                !TEST                !
!REVIEWER-CATEGORIES                !JOB-FUNCTIONS                !
!CHAPTERS                !PART-OF-MANUAL                !
!REGION-NUMBER                !REGIONS                !
!CUSTOMER-CLASS                !CUST-CLASSES                !
-----
***END***

>
```

Refer to the *SUPRA Server PDM RDM Administration Guide (VMS)*, P25-8220, for details of how to use DBAID.

Accessing Directory Views from MANTIS

From MANTIS version 2.1 and above, you can use the VIEW statement in your programs to access the Directory Views. You cannot define the user name DATABASE-DESCRIPTIONS as a MANTIS user because it is too long. Therefore, use the VIEW statement in your MANTIS programs to sign on as user name DATABASE-DESCRIPTIONS before you attempt to access a Directory View, for example:

```
VIEW ON( "DATABASE-DESCRIPTIONS" , " " )
```

Note that you must include a password null string or MANTIS will use the password specified in any preceding VIEW ON statement, or the MANTIS password if there was no preceding VIEW ON statement.

Refer to the *MANTIS Language Manual*, P25-1310, for details of how to access views from MANTIS programs.

Using Directory Views effectively

To use the Directory Views effectively, you need to know what information each view provides. The following figure shows the structure of the SUPRA Directory in terms of the Directory Views. Use this figure to ascertain which views access the information you want, then turn to the chapter describing each view for information on the view. See “[Data type descriptions](#)” on page 109 for the data type description of each column.

For example, assume you want to list all the data sets in each database description. The following figure shows that the view UX-DATA-SET-OF-DBDESC links the views UE-DATABASE-DESCRIPTION and UE-DATA-SET. “[Views accessing the relationships between entities](#),” beginning on page 67, tells you that UX-DATA-SET-OF-DBDESC contains the columns DESC and DATA-SET. “[Views accessing entity details](#),” beginning on page 29, describes the views UE-DATA-SET-OF-DBDESC and UE-DATABASE-DESCRIPTION. You design your SPECTRA process or MANTIS program to access UX-DATA-SET-OF-DBDESC using the key DESC, then map the DATA-SET column onto the key DSET to access the view UE-DATA-SET. You can thus display details of the data sets in the selected database.

2

Views accessing entity details

Entity view names start with the code UE and access details of the entities defined on the Directory. You may wish to access one entity view to obtain details of all entities of that type defined on the Directory, or you may wish to access a second entity view using columns from the first view as a key. You do this using the relationship views described in “[Views accessing the relationships between entities](#),” beginning on page 67. There are twenty entity views:

- ◆ UE-BUFFER
- ◆ UE-DATA-ITEM
- ◆ UE-DATA-SET
- ◆ UE-DATABASE-DESCRIPTION
- ◆ UE-DOMAIN
- ◆ UE-FILE-SPEC
- ◆ UE-INDEX
- ◆ UE-INDEX-FILE-SPEC
- ◆ UE-LOGICAL-DATA-ITEM
- ◆ UE-LOGICAL-VIEW
- ◆ UE-PROGRAM
- ◆ UE-RECORD
- ◆ UE-RMS-DATA-SET
- ◆ UE-RMS-FILE-SPEC
- ◆ UE-RMS-KEY
- ◆ UE-SECONDARY-KEY
- ◆ UE-SYSTEM-LOG
- ◆ UE-TASK-LOG
- ◆ UE-USER
- ◆ UE-VALIDATION-TABLE

For data type descriptions of the columns of these views, see “[Data type descriptions](#)” on page 109.

UE-BUFFER

This view accesses buffer details. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UE-BUFFER
FSI: * VSI: = MSG:          1584 BYTES USED IN OPENING VIEW.
>GO*

!          BUFF          !COPIES ! TYPE  ! SIZE  !
!-----!-----!-----!-----!
!TESTDB RBUF          !      3!     2!  1024!
!BOOKDB RBUF          !      5!     2!  6144!
!BOOKDB PBUF          !      5!     1!  1536!
!TESTDB PBUF          !      2!     1!  2048!
!TRIAL1 BUF1          !      5!     1!   512!
!-----!-----!-----!-----!
***END***
```

- ◆ BUFF is the 30-character key, constructed as follows:
 - Characters 1–6: 6-character database name
 - Characters 8–11: 4-character data set name
 - Characters 12–30: padded with spaces
- ◆ COPIES is a 2-byte integer field and indicates the number of copies of the buffer.
- ◆ TYPE is a 2-byte integer field and identifies the buffer type using the following codes:
 - 1 Primary
 - 2 Related
 - 4 RMS
- ◆ SIZE is a 2-byte integer field and indicates the size of the buffer.

UE-DATA-ITEM

This view accesses data item details. The example DBAID session shows how to open the view and retrieve the records by typing GET*.

```

>OPEN UE-DATA-ITEM
FSI: * VSI: = MSG:          2928 BYTES USED IN OPENING VIEW.
>GET*

DITM                      (+) AUBOAUTH
LENGTH                    (+) 20
USE                       (+) 2
TYPE                      (+) C
SIGN                      (+) S
DECIMAL-PLACES           (+) 0
RELATED-KEY               (+)
DISPLACEMENT              (+) 38
STAR-FILLER               (+) N
SUB-ITEMS                 (+) 0
DATA-ITEM-LEVEL          (+) 0
FSI: * VSI: + MSG: SUCCESSFUL COMPLETION

```

- ◆ DITM is the 30-character key, constructed as follows:
 - Characters 1–4: 4-character data set name
 - Characters 5–8: 4-character data item name
 - Characters 9–30: padded with spaces except for linkpath data items

If DITM is a linkpath, its key construction is:

- Characters 1–4: 4-character data set name
- Characters 5–8: 4-character data item name
- Characters 10–21: PRIMARY-DATA for a primary data set, RELATED-DATA for a linkpath in the base portions of a related data set
- Characters 22–30 : padded with spaces

If DITM is a linkpath in the coded portion of a related data set, its key construction is:

- Characters 1–4: 4-character data set name
 - Characters 5–8: 4-character data item name
 - Characters 10–19: CODED-DATA
 - Characters 21–22 : 2-character record code
 - Characters 23–30 : padded with spaces
- ◆ LENGTH is a 2-byte integer field and is the length of the data item.
 - ◆ USE is a 2-byte integer field and uses these codes to indicate data item usage:
 - 1 Root
 - 2 Key
 - 3 code
 - 4 Link
 - 5 Data
 - ◆ TYPE is a 1-character field and uses these codes to identify the data item type:
 - B Binary
 - C Character
 - F Floating point
 - K Kanji
 - N Numeric
 - P Packed numeric

- ◆ SIGN is a 1-character field and uses these codes to indicate its status:
 - S Signed (Packed numeric and Floating point data items must be signed)
 - U Unsigned
- ◆ DECIMAL-PLACES is a 2-byte integer field and indicates the number of places to the right of the decimal.
- ◆ RELATED-KEY is an 8-character field and is the referback field that refers back to a data item that is attached to a linkpath.
- ◆ DISPLACEMENT is a 2-byte integer field. Indicates how many bytes into a record the field begins, beginning with position 0.
- ◆ STAR-FILLER is a 1-character field.
- ◆ SUB-ITEMS is a 2-byte integer field and indicates the number of subdata items in the field.
- ◆ DATA-ITEM-LEVEL is a 2-byte integer field and indicates the subdata item level of the field. There are up to 9 levels.

UE-DATA-SET

This view accesses data set details. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```

>OPEN UE-DATA-SET
FSI: * VSI: = MSG:          1968 BYTES USED IN OPENING VIEW.
>GO*

!          DSET          !LENGTH ! TYPE  !CODED!STATUS!KEY-LENGTH!
!-----!-----!-----!-----!-----!-----!
!BRAN          !      156!    1!  N  ! *   !      6  !
!AUTH          !      36!    1!  N  ! *   !     20  !
!STCK          !      43!    2!  N  ! *   !      0  !
!PROD          !      78!    2!  N  ! *   !      0  !
!COMS          !     578!    2!  N  ! *   !      0  !
!ITEM          !      63!    1!  N  ! *   !      5  !
!REGN          !      42!    1!  N  ! *   !      6  !
!GILL          !       0!    1!  N  ! *   !      0  !
!REBR          !      58!    2!  N  ! *   !      0  !
!TEST          !       0!    1!  N  ! *   !      0  !
!BORE          !     124!    2!  N  ! *   !      0  !
!REVV          !      56!    1!  N  ! *   !     20  !
!AUBO          !     116!    2!  N  ! *   !      0  !
!BOOK          !      84!    1!  N  ! *   !     30  !
!RXXX          !       0!    2!  N  ! *   !      0  !
!PXXX          !      21!    1!  N  ! *   !      5  !
!CUST          !     146!    4!  N  ! *   !      0  !
!-----!-----!-----!-----!-----!
***END***

```

- ◆ DSET is the 30-character key, 4-character data set name, padded to the right with spaces to 30 characters.
- ◆ LENGTH is a 2-byte integer field and is the record length of the data set in bytes.
- ◆ TYPE is a 2-byte integer field and identifies the data set type using the following codes:
 - 1 Primary
 - 2 Related
 - 4 RMS

- ◆ CODED is a 1-character field that indicates whether the data set contains coded records by using the following codes:
 - Y Coded
 - N Not coded
- ◆ STATUS is a 1-character field and uses the following codes to show the status of the data set:
 - * OK
 - C Being compiled
 - V Being validated
 - N Needs validation
 - M Being modified
 - + Validated OK
 - P Being printed
- ◆ KEY-LENGTH is a 2-byte integer field and is the length of the key of the data set.

UE-DATABASE-DESCRIPTION

This view accesses database details. The example DBAID session shows how to open the view and retrieve the records by typing GET*.

```

>OPEN UE-DATABASE-DESCRIPTION
FSI: *   VSI: =   MSG:           3504 BYTES USED IN OPENING VIEW.
>GET*

DESC                (+) TESTDB
PASSWORD            (+)
MAX-HELD-RECORDS    (+) 16
MAX-TASKS           (+) 2
MAX-UPDATE-TASKS    (+) 2
SHADOW-OPTION       (+) N
SINGLE-TASK          (+) N
STATUS              (+) N
DATE-COMPILED       (+) 881031
TIME-COMPILED       (+) 16202795
ACCESS-METHOD      (+) Q
GLOBAL-SECT-TYPE    (+) G
CALLING-MECH        (+) R
CLUSTER-NETWORK     (+) L
FSI: *   VSI: +   MSG: SUCCESSFUL COMPLETION

```

For more information about these fields, refer to the *SUPRA Server PDM Database Administration Guide (UNIX & VMS)*, P25-2260.

- ◆ DESC is the 30-character key, 6-character database description name, padded to the right with spaces to 30 characters.
- ◆ PASSWORD is an 8-character field.
- ◆ MAX-HELD-RECORDS is a 2-byte integer field.
- ◆ MAX-TASKS is a 2-byte integer field.
- ◆ MAX-UPDATE-TASKS is a 2-byte integer field.

- ◆ SHADOW-OPTION is a 1-character field and identifies how shadow files will be used in the event of a read/write failure:
 - N No shadow recording
 - C Continue processing using the shadow file
- ◆ SINGLE-TASK is a 1-character field and identifies the database as single-task or multi-user:
 - Y Single-task database
 - N Multi-user database
- ◆ STATUS is a 1-character field and uses the following codes to show the status of the database:
 - * OK
 - C Being compiled
 - V Being validated
 - N Needs validation
 - M Being modified
 - + Validated OK
 - P Being printed
- ◆ DATE-COMPILED is a 4-byte integer field.
- ◆ TIME-COMPILED is a 4-byte integer field.
- ◆ ACCESS-METHOD is a 1-character field and uses the following codes to indicate the method used by the PDM for database I/O:
 - R RMS
 - Q QIO

- ◆ GLOBAL-SECT-TYPE is a 1-character field and uses the following codes to indicate the global section type used to access the database description module:
 - G Groupwide global sections
 - S Systemwide global sections
- ◆ CALLING-MECH is a 1-character field and uses these codes to identify the method used to pass parameters to the PDM:
 - R Reference only
 - D By reference and descriptor
- ◆ CLUSTER-NETWORK is a 1-character field and uses these codes to indicate whether this database will run in a clustered environment:
 - C Cluster/network support
 - L Local

UE-DOMAIN

This view accesses domain details. The example DBAID session shows how to open the view and retrieve the records by typing GET*.

```

>OPEN UE-DOMAIN
FSI: * VSI: = MSG: 4888 BYTES USED IN OPENING VIEW.
>GET*

DOMN (+) DELIVERY-ROUTES
FUNCTION (+) 7
UNIT (+) 3500
FORMAT (+) C
LENGTH (+) 10
DEC-PLACES (+) 0
SIGN-OPT (+) S
NULLS-ALLOWED (+) Y
NULL-VALUE (+) Unassigned
DEFAULT (+) Railways
GET-VAL-OPT (+) N
VAL-OPT (+) N
VAL-MIN (+)
VAL-MAX (+)
VAL-EXIT (+)
STATUS (+) *
FSI: * VSI: + MSG: SUCCESSFUL COMPLETION

```

- ◆ DOMN is the 30-character key, domain name, padded to the right with spaces to 30 characters.

- ◆ FUNCTION is a 2-byte integer field and uses these codes to identify the category of data described by the domain:
 - 1 Area
 - 2 Date
 - 3 Distance
 - 4 Money
 - 5 Number
 - 6 Pressure
 - 7 String
 - 8 Temperature
 - 9 Time
 - 10 Velocity
 - 11 Volume
 - 12 Weight

- ◆ UNIT is a 2-byte integer field and identifies the unit in which the category of data in the domain is measured. Refer to the *SUPRA Server PDM Database Administration Guide (UNIX & VMS)*, P25-2260, for the DOMAIN-UNIT translate table.

- ◆ FORMAT is a 1-character field and uses these codes to indicate the data type:
 - 1 Primary
 - 2 Related
 - 4 RMS

- ◆ LENGTH is a 4-byte integer field and is the length of the data items in the domain. This value must remain constant; there can be only one length for all data items.

- ◆ DEC-PLACES is a 4-byte integer field and is the number of decimal places to the right of the decimal.
- ◆ SIGN-OPT is a 1-character field and contains S if the data item is signed, and U if it is unsigned.
- ◆ NULLS ALLOWED is a 1-character field, and uses the following options to specify whether nulls are allowed:
 - Y Nulls are allowed
 - N Nulls are not allowed
- ◆ NULL-VALUE is a 32-character field and is the value that represents Null for this data item. This field is only necessary if Nulls are allowed.
- ◆ DEFAULT is a 32-character field.
- ◆ GET-VAL-OPT is a 1-character field and uses the following options to specify whether RDM performs validation during a GET:
 - Y RDM performs validation during a GET
 - N RDM does not perform validation during a GET
- ◆ VAL-OPT is a 1-character field and uses these codes to indicate the type of validation performed:
 - N No validation
 - R Range of values
 - T Validation table
 - E User defined validation exit
- ◆ VAL-MIN is a 32-character field and is the minimum value of the range of data stored in this domain.
- ◆ VAL-MAX is a 32-character field and is the maximum value of the range of data stored in this domain.

- ◆ VAL-EXIT is an 8-character field and is the exit name. Refer to the *SUPRA Server PDM RDM Administration Guide (VMS)*, P25-8220, for information on validation exits.
- ◆ STATUS is a 1-character field and uses these codes to show the status of the domain:
 - * OK
 - C Being compiled
 - V Being validated
 - N Needs validation
 - M Being modified
 - + Validated OK
 - P Being printed

UE-FILE-SPEC

This view accesses file specification details. The example DBAID session shows how to open the view and retrieve the records using a keyed GET.

```

>OPEN UE-FILE-SPEC
FSI: * VSI: = MSG:          8184 BYTES USED IN OPENING VIEW.
>GET* USING 'REVW BOOKDB FILE-SPEC 01'

FSPC                      (+) REVW BOOKDB FILE-SPEC 01
TOTAL-RECORDS              (+) 36
RECS-PER-BLOCK             (+) 18
CONTROL-INTERVAL          (+) 100
LOAD-LIMIT                 (+) 80
ALLOCATION-1                (+) 1
ALLOCATION-2                (+) 0
ALLOCATION-3                (+) 0
ALLOCATION-4                (+) 0
FSPEC-1                    (+) DB:REVW.NEW
SHADOW-FSPEC-1            (-)
FSPEC-2                    (-)
SHADOW-FSPEC-2            (-)
FSPEC-3                    (-)
SHADOW-FSPEC-3            (-)
FSPEC-4                    (-)
SHADOW-FSPEC-4            (-)
FSI: * VSI: + MSG: SUCCESSFUL COMPLETION

```

- ◆ FSPC is the 30-character key, constructed as follows:
 - Characters 1–4: 4-character data set name
 - Characters 6–11: 6-character database name
 - Characters 13–21: FILE-SPEC
 - Characters 23–24: 01 for allocation 1
02 for allocation 2
03 for allocation 3
04 for allocation 4
 - Characters 25–30: padded with spaces
- ◆ TOTAL-RECORDS is a 4-byte integer field.

- ◆ RECS-PER-BLOCK is a 2-byte integer field.
- ◆ CONTROL-INTERVAL is a 4-byte integer field.
- ◆ LOAD-LIMIT is a 2-byte integer field.
- ◆ ALLOCATION-1 is a 4-byte integer field and is the percentage of records in FSPEC-1 and SHADOW-FSPEC-1 if used.
- ◆ ALLOCATION-2 is a 4-byte integer field and is the percentage of records in FSPEC-2 and SHADOW-FSPEC-2 if used.
- ◆ ALLOCATION-3 is a 4-byte integer field and is the percentage of records in FSPEC-3 and SHADOW-FSPEC-3 if used.
- ◆ ALLOCATION-4 is a 4-byte integer field and is the percentage of records in FSPEC-4 and SHADOW-FSPEC-4 if used.
- ◆ FSPEC-1 is a 44-character field that displays the full VMS path specification of the allocation-1 part of the data set.
- ◆ SHADOW-FSPEC-1 is a 44-character field that displays the full VMS path specification of the shadow file of FSPEC-1 if applicable.
- ◆ FSPEC-2 is a 44-character field that displays the full VMS path specification of the allocation-2 part of the data set.
- ◆ SHADOW-FSPEC-2 is a 44-character field that displays the full VMS path specification of the shadow file of FSPEC-2 if applicable.
- ◆ FSPEC-3 is a 44-character field that displays the full VMS path specification of the allocation-3 part of the data set.
- ◆ SHADOW-FSPEC-3 is a 44-character field that displays the full VMS path specification of the shadow file of FSPEC-3 if applicable.
- ◆ FSPEC-4 is a 44-character field that displays the full VMS path specification of the allocation-4 part of the data set.
- ◆ SHADOW-FSPEC-4 is a 44-character field that displays the full VMS path specification of the shadow file of FSPEC-4 if applicable.

UE-INDEX

This view accesses index details. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```

>OPEN UE-INDEX
FSI: * VSI: = MSG:          1200 BYTES USED IN OPENING VIEW.
>GO*

!           INDX           !STATUS!
!-----!-----!
!BOOKIXPU           !           !
!PRODIXPP           !           !
-----
***END***

```

- ◆ INDX is the 30-character key, constructed as follows:
 - Characters 1–4: 4-character data set name
 - Characters 5–6: IX indicating that this is an index
 - Characters 7–8: 2-character index name
 - Characters 9–30: padded with spaces
- ◆ STATUS is a 1-character field and is reserved for future use.

UE-INDEX-FILE-SPEC

This view accesses details of the physical file attributes for a specified index. The example DBAID session shows how to open the view and retrieve the records by typing GET*.

```

>OPEN UE-INDEX-FILE-SPEC
FSI: * VSI: = MSG:          2968 BYTES USED IN OPENING VIEW.
>GET*

FSPC                      (+) PRODIXPP  BOOKDB FILE-SPEC 01
CORRUPT-ACTION             (+) O
NULL-SORTING               (+) H
READ-VERIFY                (+) Y
FILE-SPEC                  (+) REL11DATA:PRODIXPP.IDX
SHADOW-FILE-SPEC          (-)
FSI: * VSI: - MSG: SUCCESSFUL COMPLETION

```

- ◆ FSPC is the 30-character key, constructed as follows:
 - Characters 1–4: 4-character data set name
 - Characters 5–6: IX indicating this is an index
 - Characters 7–8: 2-character index name
 - Characters 12–17: 6-character database name
 - Characters 19–27: FILE-SPEC
 - Characters 29–30: 01 for allocation 1
02 for allocation 2
03 for allocation 3
04 for allocation 4

- ◆ CORRUPT-ACTION is a 1-character field and uses one of the following single character codes to indicate the action the PDM will take if the index file becomes corrupted:
 - O Operator; marks the index as unavailable and prompts the user at the operator console to either continue without using the index or to perform a dynamic populate.
 - C Continue; marks the index as unavailable and continues processing without using the corrupt index file.
 - P Populate; performs a dynamic populate on the corrupt index file.

- ◆ NULL-SORTING is a 1-character field and uses these codes to identify where nulls are sorted in the collating sequence:
 - H High
 - L Low

- ◆ READ-VERIFY is a 1-character field and uses the following codes to specify whether the PDM checks for a corrupt index when reading an index file:
 - Y Yes
 - N No

- ◆ FILE-SPEC is a 44-character field and displays the full VMS path specification of the index file. This file specification can be a logical name or include a logical name.

- ◆ SHADOW-FILE-SPEC is a 44-character field and displays the full VMS path specification of the shadow index file, if there is one. This file specification can be a logical name or include a logical name.

UE-LOGICAL-DATA-ITEM

This view accesses logical data item names. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UE-LOGICAL-DATA-ITEM
FSI: * VSI: = MSG: 1008 BYTES USED IN OPENING VIEW.
>GO*

!           LGDI           !
!-----!
!COMMENT-1           !
!BRANCH-SLS-QUOTA           !
!CUSTOMER-ZIPCODE           !
!START-WRITING           !
!CUSTOMER-SURNAME           !
!CUSTOMER-CHRISTIAN-NAME           !
!CUSTOMER-CR-LIM           !
!REGION-TO-BRANCH           !
!COMMENT-9           !
!MANUAL-IN-PRODUCTION           !
!REVIEW-SUBMIT           !
!CHRISTIAN-NAME           !
!BRANCH-ADDR           !
!PRODUCT-GROUP           !
!BRANCH-NO           !
!CUSTOMER-NO           !
!MANUAL           !
!-----!
***MORE***
```

LGDI is the 30-character key, logical data item name, padded to the right with spaces to 30 characters.

UE-LOGICAL-VIEW

This view accesses view details. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```

>OPEN UE-LOGICAL-VIEW
FSI: * VSI: = MSG:          1968 BYTES USED IN OPENING VIEW.
>GO*

!                LV                !STATUS!BOUND!  DATE  !LV-BOUND-SIZE! TIME !
!-----!-----!-----!-----!-----!-----!
!REGION                ! * ! Y !19871020!          952 !100722!
!CUSTOMER              ! * ! Y !19871014!         1776 !152353!
!REVIEWERS             ! * ! N !19870526!           0 !135503!
!PRODUCTION-DETAILS   ! * ! N !19870602!           0 !155430!
!PRINTED-MANUALS      ! * ! N !19870602!           0 !155807!
!INDEXED-BRANCHES-IN-REGION ! * ! N !19871014!           0 !121005!
!INDEXED-REGION       ! * ! N !19871014!           0 !120448!
!REVIEW-DETAILS       ! * ! N !19870527!           0 !104806!
!BASE-VIEW-B          ! * ! N !19871125!           0 !121914!
!MANUAL-COMMENTS      ! * ! N !19870526!           0 !110537!
!PRODUCTS-IN-REGION   ! * ! N !19871014!           0 !130103!
!DEVELOPERS           ! * ! N !19881219!           0 !155349!
!UPDATE-EXISTING-MANUAL ! * ! N !19880210!           0 !162533!
!INDEXED-CUSTOMER     ! * ! N !19871014!           0 !120645!
!BRANCH               ! * ! N !19871020!           0 !100711!
!INDEXED-BRANCH-SUBSET ! * ! N !19871014!           0 !120844!
!BASE-VIEW-A          ! * ! N !19871125!           0 !121608!
!-----!-----!-----!-----!-----!
***MORE***

```

- ◆ LV is the 30-character key, view name, padded to the right with spaces to 30 characters.
- ◆ STATUS is a 1-character field and uses these codes to show the status of the view:
 - * OK
 - C Being compiled
 - V Being validated
 - N Needs validation
 - M Being modified
 - + Validated OK
 - P Being printed
- ◆ BOUND is a 1-character field and uses the following options to indicate whether the view is bound:
 - Y Bound
 - N Not bound
- ◆ DATE is an 8-character field.
- ◆ LV-BOUND-SIZE is a 4-byte integer field and is the size in bytes of the bound view.
- ◆ TIME is an 8-character field.

UE-PROGRAM

This view accesses program details. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```

>OPEN UE-PROGRAM
FSI: * VSI: = MSG:          1968 BYTES USED IN OPENING VIEW.
>GO*

!          PROG          !STATUS!DATE-ENR!TIME-E!MOD-LEV!LANGUA!
!-----!-----!-----!-----!-----!-----!
!GET-MANUALS          ! * !19890410!161901!          7! C !
!MANUALS          ! * !19890410!160109!          4! C !
!-----!-----!-----!-----!-----!-----!
***END***

```

- ◆ PROG is the 30-character key, program name, padded to the right with spaces to 30 characters.
- ◆ STATUS is a 1-character field and uses these codes to show the status of the data set:
 - * OK
 - C Being compiled
 - V Being validated
 - N Needs validation
 - M Being modified
 - + Validated OK
 - P Being printed

- ◆ DATE-ENROLLED is an 8-character field and is the data the program was precompiled and entered into the Directory.
- ◆ TIME-ENROLLED is an 8-character field and is the time the program was precompiled and entered into the Directory.
- ◆ MOD-LEVEL is a 2-byte integer field and is the modification level. The first time a program is precompiled and enrolled, MOD-LEVEL is set to 0. Each subsequent precompile and enrollment increments the MOD-LEVEL. Refer to the *SUPRA Server PDM Database Administration Guide (UNIX & VMS)*, P25-2260, for details.
- ◆ LANGUAGE is a one character field and uses these codes to indicate the language in which the program is written:
 - C COBOL
 - F FORTRAN
 - B BASIC

UE-RECORD

This view accesses details of physical records. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```

>OPEN UE-RECORD
FSI: *   VSI: =   MSG:           1968 BYTES USED IN OPENING VIEW.
>GO*

!          RECD          ! TYPE  ! CODE !NO-OF-D!LENGTH !REDEFINE!
!-----!-----!-----!-----!-----!-----!
!PXXX PRIMARY-DATA      !      1!  ** !      2!   21!      !
!BORE RELATED-DATA      !      2!  ** !      9!  124!      !
!COMS RELATED-DATA      !      2!  ** !     11!  578!      !
!REGN PRIMARY-DATA      !      1!  ** !      4!   42!      !
!ITEM PRIMARY-DATA      !      1!  ** !      8!   63!      !
!REVV PRIMARY-DATA      !      1!  ** !      5!   56!      !
!RXXX RELATED-DATA      !      2!  ** !      2!    0!      !
!REBR RELATED-DATA      !      2!  ** !      5!   58!      !
!AUTH PRIMARY-DATA      !      1!  ** !      3!   36!      !
!PROD RELATED-DATA      !      2!  ** !      6!   78!      !
!GILL PRIMARY-DATA      !      1!  ** !      3!    0!      !
!CUST RMS-DATA          !      4!  ** !     10!  146!      !
!TEST PRIMARY-DATA      !      1!  ** !      3!    0!      !
!STCK RELATED-DATA      !      2!  ** !      7!   43!      !
!BRAN PRIMARY-DATA      !      1!  ** !     13!  156!      !
!BOOK PRIMARY-DATA      !      1!  ** !      8!   84!      !
!AUBO RELATED-DATA      !      2!  ** !      8!  116!      !
!-----!-----!-----!-----!-----!
***END***

```

- ◆ RECD is the 30-character key, constructed as follows:
 - Characters 1–4: 4-character data set name
 - Characters 6–17: PRIMARY-DATA for a primary data set
RELATED-DATA for the base portion of a related data set
RMS-DATA (4 spaces) for an RMS data set
 - Characters 18–30: padded with spaces

If RECD is a coded record, its key construction is:

- Characters 1–4: 4-character data set name
 - Characters 6–17: CODED-DATA
 - Characters 17–18: 2-character record code
 - Characters 19–30: padded with spaces
- ◆ TYPE is a 2-byte integer field and uses these codes to identify the record type:
 - 1 Primary
 - 2 Related
 - 4 RMS
 - ◆ CODE is a 2-character field and contains the record code that redefines record data in the same file.
 - ◆ NO-OF-DATA-ITEMS is a 2-byte integer field.
 - ◆ LENGTH is a 2-byte integer field and is the length of the record.
 - ◆ REDEFINED-01 is an 8-character field.

UE-RMS-DATA-SET

This view accesses data set details, returning data for both PDM data sets and RMS data sets. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UE-RMS-DATA-SET
FSI: *   VSI: =   MSG:           1584 BYTES USED IN OPENING VIEW.
>GO*

!           DSET                !LENGTH ! TYPE  !STATUS!
!-----!-----!-----!-----!
!BRAN                !   156!    1!   *   !
!AUTH                !   36!    1!   *   !
!STCK                !   43!    2!   *   !
!PROD                !   78!    2!   *   !
!COMS                !  578!    2!   *   !
!ITEM                !   63!    1!   *   !
!REGN                !   42!    1!   *   !
!REBR                !   58!    2!   *   !
!TEST                !    0!    1!   *   !
!BORE                !  124!    2!   *   !
!REVV                !   56!    1!   *   !
!AUBO                !  116!    2!   *   !
!BOOK                !   84!    1!   *   !
!RXXX                !    0!    2!   *   !
!PXXX                !   21!    1!   *   !
!CUST                !  146!    4!   *   !
!-----!-----!-----!-----!
***END***
```

- ◆ DSET is the 30-character key, 4-character data set name, padded to the right with spaces to 30 characters.
- ◆ LENGTH is a 2-byte integer field and is the length of the record in the data set.
- ◆ TYPE is a 2-byte integer field and identifies the data set type using the following codes:
 - 1 Primary
 - 2 Related
 - 4 RMS

- ◆ STATUS is a 1-character field and uses these codes to show the status of the data set:
 - * OK
 - C Being compiled
 - V Being validated
 - N Needs validation
 - M Being modified
 - + Validated OK
 - P Being printed

UE-RMS-FILE-SPEC

This view accesses file specification details for both RMS and PDM data sets. The example DBAID session shows how to open the view and retrieve the records by typing GET*.

```
>OPEN UE-RMS-FILE-SPEC
FSI: * VSI: = MSG:          1904 BYTES USED IN OPENING VIEW.
>GET*
FSPC                                (+) AUTH BOOKDB FILE-SPEC 01
BUCKET-SIZE                         (+) 0
FSPEC-1                             (+) DB:AUTH.INF
FSI: * VSI: + MSG: SUCCESSFUL COMPLETION
```

- ◆ FSPC is the 30-character key, constructed as follows:
 - Characters 1–4: 4-character data set name
 - Characters 6–11: 6-character database name
 - Characters 13–21: FILE-SPEC
 - Characters 23–24: 01 for allocation 1
02 for allocation 2
03 for allocation 3
04 for allocation 4
 - Characters 25–30: padded with spaces
- ◆ BUCKET-SIZE is a 2-byte integer field.
- ◆ FSPEC-1 is a 44-character field that displays the full VMS path specification of the data set.
- ◆ This view retrieves similar information as the view UE-FILE-SPEC; however, the field BUCKET-SIZE overlays the field ALLOCATION-4 in the view UE-FILE-SPEC.

UE-RMS-KEY

This view accesses RMS key details. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UE-RMS-KEY
FSI: * VSI: = MSG:          1776 BYTES USED IN OPENING VIEW.
>GO*

!          RMSK          !  NUMBER  !  LENGTH  !UNIQUE!CAN-BE!
!-----!-----!-----!-----!-----!
!CUSTNAMEKEY          !          1!          40!  Y  !  Y  !
!CUSTBASEKEY          !          0!          3!  Y  !  N  !
!-----!-----!-----!-----!-----!
***END***
```

- ◆ RMSK is the 30-character key, RMS key name, padded to the right with spaces to 30 characters. The RMS key name is 12 characters long, and is constructed as follows:
 - Characters 1–4: name of the data set
 - Characters 5–12: name of the RMS key as specified through DBA
 - Characters 13–30: padded with spaces
- ◆ NUMBER is a 4-byte integer and is the RMS key number. 0 is a primary key; alternative keys may have a number between 1 and 254.
- ◆ LENGTH is a 4-byte integer and is the key length.
- ◆ UNIQUE-KEY is a 1-character field and uses the following options to indicate whether the key is unique:
 - Y Unique
 - N Not unique
- ◆ CAN-BE-CHANGED is a 1-character field.

UE-SECONDARY-KEY

This view accesses secondary key details. The example DBAID session shows how to open the view and retrieve the records by typing GET*.

```

>OPEN UE-SECONDARY-KEY
FSI: *   VSI: =   MSG:           2352 BYTES USED IN OPENING VIEW.
>GET*

SKEY                (+) BOOKSKNO
UNIQUE-KEY          (+) N
KEY-DIRECTION       (+) F
POINTER-ORDERING   (+) N
POINTER-TYPE        (+) D
DATA-TYPE-SORT      (+) N
DUPL-ALLOWED       (+) 5
STATUS              (+) N
FSI: *   VSI: +   MSG: SUCCESSFUL COMPLETION

```

- ◆ SKEY is the 30-character key, constructed as follows:
 - Characters 1–4: 4-character data set name
 - Characters 5–6: SK indicating that this is a secondary key
 - Characters 7–8: 2-character key name
 - Characters 9–30: padded with spaces
- ◆ UNIQUE-KEY is a 1-character field and uses these codes to specify whether the index supports duplicate secondary keys:
 - Y Duplicate keys supported
 - N No duplicate keys allowed

- ◆ KEY-DIRECTION is a 1-character field and uses these codes to indicate the direction in which the keys are sorted in the file:
 - F Forward (ascending order)
 - R Reverse (descending order)
 - B Both (ascending and descending)
- ◆ POINTER-ORDERING is a 1-character field and uses these codes to indicate whether the PDM uses pointers. The PDM uses pointers to ensure that it retrieves records with identical keys in the order in which they occur in the data set file:
 - Y Ordered
 - N Not ordered
- ◆ POINTER-TYPE is a 1-character field and uses these codes to identify the pointer type stored with the secondary key:
 - D Direct (using Relative Record Number, RRN)
 - I Indirect (using a control key)
- ◆ DATA-TYPE-SORT is a 1-character field and uses the following codes to specify whether the secondary keys are sorted according to data type or treated as character strings:
 - Y Sorted by data type
 - N Sorted as character strings
- ◆ DUPL-ALLOWED is a 2-byte integer field and identifies the percentage of duplicates expected for this secondary key.
- ◆ STATUS is a 1-character field and is reserved for future use.

UE-SYSTEM-LOG

This view accesses system log details. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```

>OPEN UE-SYSTEM-LOG
FSI: * VSI: = MSG:          4128 BYTES USED IN OPENING VIEW.
>GO*

!                FLOG                !BLOCK-SIZE!NO-OF-BLOCKS!
!-----!-----!-----!
!TESTDB FUNCTION-LOG                !      1 !           500!
=====
!                FSPEC-1                !
!                FSPEC-2                !
!                SHADOW-FSPEC-1          !
!                SHADOW-FSPEC-2          !
-----
!DB:TESTDBSYSLOG1.LOG                !
!DB:TESTDBSYSLOG2.LOG                !
!                                     !
!                                     !
-----
***MORE***

```

- ◆ FLOG is the 30-character key, system log, constructed as follows:
 - Characters 1–6: 6-character database name
 - Characters 8–19: FUNCTION-LOG or SYSTEM-LOG
 - Characters 20–30: padded with spaces
- ◆ BLOCK-SIZE is a 2-byte integer field.
- ◆ NO-OF-BLOCKS is a 4-byte integer field.
- ◆ FSPEC-1 is a 44-character field that displays the full VMS path specification of the first component of the system log.
- ◆ FSPEC-2 is a 44-character field that displays the full VMS path specification of the second component of the system log.
- ◆ SHADOW FSPEC-1 is a 44-character field that displays the full VMS path specification of the shadow file of the first component of the system log if applicable.
- ◆ SHADOW-FSPEC-2 is a 44-character field that displays the full VMS path specification of the shadow file of the second component of the system log if applicable.

UE-TASK-LOG

This view accesses task log details. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```

>OPEN UE-TASK-LOG
FSI: * VSI: = MSG:          2968 BYTES USED IN OPENING VIEW.
>GO*

!          TLOG          !BLOCK-SIZE!NO-OF-BLOCKS!NO-OF-BUFFERS!
!-----!-----!-----!-----!
!TESTDB TASK-LOG          !      1  !          20!          3  !
=====
!          FSPEC-1          !
!          SHADOW-FSPEC-1          !
-----
!DB:TESTDB-TASKLOG.LOG          !
!          !
-----
***MORE***

```

- ◆ TLOG is the 30-character key, task log, constructed as follows:
 - Characters 1–6: 6-character database name
 - Characters 8–15: TASK-LOG
 - Characters 16–30: padded with spaces
- ◆ BLOCK-SIZE is a 2-byte integer field.
- ◆ NO-OF-BLOCKS is a 4-byte integer field.
- ◆ NO-OF-BUFFERS is a 2-byte integer field.
- ◆ FSPEC-1 is a 44-character field that displays the full VMS path specification of the task log file.
- ◆ SHADOW-FSPEC-1 is a 44-character field that displays the full VMS path specification of the shadow task log file if applicable.

UE-USER

This view accesses user details. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```

>OPEN UE-USER
FSI: * VSI: = MSG:          1392 BYTES USED IN OPENING VIEW.
>GO*

!          USER          !AUTHORITY!PASSWORD!
!-----!-----!-----!
!DA          !          30 !â S {£JY!
!FRED        !          30 !â 3 X>+ö!
!DATA-DICTIONARY !          20 !â S {£JY!
!CLOUGH      !          20 !â S {£JY!
!SSSS        !          40 !â S {£JY!
!DATABASE-DESCRIPTIONS !          20 !â S {£JY!
!PROGRAMMER  !          60 !â S {£JY!
!CINCOM      !          0 !âeßp° !
!DP          !          40 !â S {£JY!
!PUBLIC      !          50 !âD;ê+tiO!
!AAAA       !          30 !â S {£JY!
-----
***END***

```

- ◆ USER is the 30-character key, username, padded to the right with spaces to 30 characters.
- ◆ AUTHORITY is a 2-byte integer field and uses these codes to show the access authority of the user name:
 - 10 Cincom
 - 20 Privileged
 - 30 DBA/Utilities
 - 40 Development personnel
 - 50 Read-only
 - 60 RDM user
- ◆ PASSWORD is an 8-character field and is encrypted.

UE-VALIDATION-TABLE

This view accesses validation table details. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UE-VALIDATION-TABLE
FSI: * VSI: = MSG:          1200 BYTES USED IN OPENING VIEW.
>GO*

!           VTBL           !STATUS!
!-----!-----!
!REGIONS           !           !
!PART-OF-MANUAL    !           !
!JOB-FUNCTIONS     !           !
!CUST-CLASSES      !           !
!TEST              !           !
-----
***END***
```

- ◆ VTBL is the 30-character key, validation table name, padded to the right with spaces to 30 characters.
- ◆ STATUS is a 1-character field and is reserved for future use.

3

Views accessing the relationships between entities

Directory Views that provide data about the connection between Directory entities start with the letters UX. They enable you to access one entity view from another entity view, using one or more columns as the key. See [“Using Directory Views effectively”](#) on page 27 for details of how to access a UE view from a UX view.

The following Directory Views provide information about relationships between entities:

UX-BUFFER-OF-DATA-SET	UX-LOGICAL-DATA-ITEM-OF-LV
UX-BUFFER-OF-DBDESC	UX-LOGICAL-VIEW-OF-DBDESC
UX-DATA-ITEM-IN-SEC-KEY	UX-PHYSICAL-TO-LOGICAL-DATA-IT
UX-DATA-ITEM-OF-RECORD	UX-PROGRAM-USES-LOGICAL-VIEW
UX-DATA-SET-OF-DBDESC	UX-RECORD-OF-DATA-SET
UX-DERIVED-VIEW-CONSTANT	UX-RMS-KEY-IN-DATA-ITEM
UX-DOMAIN-OF-DATA-ITEM	UX-RMS-KEY-IN-DATA-SET
UX-DOMAIN-USES-VALIDATION-TABL	UX-SEC-KEY-IN-INDEX
UX-FILE-SPEC-OF-DATA-SET	UX-SUB-DATA-ITEM
UX-FILE-SPEC-OF-DATABASE	UX-SYSTEM-LOG-OF-DBDESC
UX-FILE-SPEC-OF-INDEX	UX-TASK-LOG-OF-DBDESC
UX-INDEX-IN-DATA-SET	UX-USER-MAY-RUN-PROGRAM
UX-INDEX-OF-DBDESC	UX-USER-MAY-USE-LOGICAL
UX-LINKPATH-OF-RECORD	UX-VIEW-TO-VIEW
UX-LOGICAL-DATA-ITEM-OF-DATA-S	

For data type descriptions of the columns of these views, see [“Data type descriptions”](#) on page 109.

UX-BUFFER-OF-DATA-SET

This view connects the views UE-DATA-SET and UE-BUFFER. Use it to provide buffer details for a specified data set. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```

>OPEN UX-BUFFER-OF-DATA-SET
FSI: * VSI: = MSG:          2544 BYTES USED IN OPENING VIEW.
>GO*

!           DSET           !           BUFFER           !
!-----!-----!-----!
!BRAN          !TESTDB PBUF          !
!AUTH          !BOOKDB PBUF          !
!STCK          !TESTDB RBUF          !
!PROD          !BOOKDB RBUF          !
!COMS          !BOOKDB RBUF          !
!ITEM          !TESTDB PBUF          !
!REGN          !TESTDB PBUF          !
!REBR          !TESTDB RBUF          !
!BORE          !BOOKDB RBUF          !
!REVV          !BOOKDB PBUF          !
!AUBO          !BOOKDB RBUF          !
!BOOK          !BOOKDB PBUF          !
!PXXX          !TRIAL1 BUF1          !
!-----!-----!-----!
***END***
    
```

- ◆ DSET is the 30-character key, 4-character data set name, padded to the right with spaces to 30 characters.
- ◆ BUFFER is the 30-character column that you can use to access the view UE-BUFFER. BUFFER contains the following:
 - Characters 1–6: 6-character database name
 - Characters 8–11: 4-character buffer name
 - Characters 12–30: padded with spaces

UX-BUFFER-OF-DBDESC

This view connects the views UE-DATABASE-DESCRIPTION and UE-BUFFER. Use it to provide buffer details for a specified database. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UX-BUFFER-OF-DBDESC
FSI: *   VSI: =   MSG:          2544 BYTES USED IN OPENING VIEW.
>GO*

!          DESC          !          BUFFER          !
!-----!-----!-----!
!TESTDB          !TESTDB PBUF          !
!TESTDB          !TESTDB RBUF          !
!SPARED          !SPARED PBUF          !
!SPARED          !SPARED RBUF          !
!BOOKDB          !BOOKDB PBUF          !
!BOOKDB          !BOOKDB RBUF          !
!TRIAL1          !TRIAL1 BUF1          !
!-----!-----!-----!
***END***
```

- ◆ DESC is the 30-character key, 6-character data set name, padded to the right with spaces to 30 characters.
- ◆ BUFFER is the 30-character column that you can use to access the view UE-BUFFER. BUFFER contains the following:
 - Characters 1–6: 6 character database name
 - Characters 8–11: 4-character buffer name
 - Characters 12–30: padded with spaces

UX-DATA-ITEM-IN-SEC-KEY

This view connects the views UE-SECONDARY-KEY and UE-DATA-ITEM. Use it to provide details of the data items in a specified secondary key. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UX-DATA-ITEM-IN-SEC-KEY
FSI: * VSI: = MSG:          2544 BYTES USED IN OPENING VIEW.
>GO*

!                SKEY                !                DATA-ITEM                !
!-----!-----!-----!-----!
!PRODSKNO                !PRODPRNT                !
!BOOKSKNO                !BOOKPUBL                !
!-----!-----!-----!-----!
***END***
```

- ◆ SKEY is the 30-character key, constructed as follows:
 - Characters 1–4: 4-character data set name
 - Characters 5–6: SK indicating that this is a secondary key
 - Characters 7–8: 2-character key name
 - Characters 9–30: padded with spaces
- ◆ DATA-ITEM is the 30-character column that you can use to access the view UE-DATA-ITEM. DATA-ITEM contains the following:
 - Characters 1–4: 4-character data set name
 - Characters 5–8: 4-character data item name
 - Characters 10–30: padded with spaces

UX-DATA-ITEM-OF-RECORD

This view connects the views UE-RECORD and UE-DATA-ITEM. Use it to provide details of the data items in a specified record. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```

>OPEN UX-DATA-ITEM-OF-RECORD
FSI: *   VSI: =   MSG:           2544 BYTES USED IN OPENING VIEW.
>GO*

!                RECD                !                DATA-ITEM                !
!-----!-----!-----!-----!
!PXXX PRIMARY-DATA                !PXXXROOT                !
!PXXX PRIMARY-DATA                !PXXXCTRL                !
!BORE RELATED-DATA                !BOREBOOK                !
!BORE RELATED-DATA                !BOOKLKRE RELATED-DATA  !
!BORE RELATED-DATA                !BOREREVW                !
!BORE RELATED-DATA                !REVWLKBO RELATED-DATA  !
!BORE RELATED-DATA                !BORESDAT                !
!BORE RELATED-DATA                !BOREDDAT                !
!BORE RELATED-DATA                !BOREADAT                !
!BORE RELATED-DATA                !BORETASK                !
!BORE RELATED-DATA                !BORECHAP                !
!COMS RELATED-DATA                !COMSBOOK                !
!COMS RELATED-DATA                !BOOKLKCO RELATED-DATA  !
!COMS RELATED-DATA                !COMSCOM1                !
!COMS RELATED-DATA                !COMSCOM2                !
!COMS RELATED-DATA                !COMSCOM3                !
!COMS RELATED-DATA                !COMSCOM4                !
!-----!-----!-----!-----!
***MORE***

```

- ◆ RECD is the 30-character key, constructed as follows:
 - Characters 1–4: 4-character data set name
 - Characters 6–17: PRIMARY-DATA for a primary data set
RELATED-DATA for the base portion of a related data set
RMS-DATA (4 spaces) for an RMS data set
 - Characters 18–30: padded with spaces

If RECD is a coded record, its key construction is:

- Characters 1–4: 4-character data set name
 - Characters 6–17: CODED-DATA
 - Characters 17–18: 2-character record code
 - Characters 19–30: padded with spaces
- ◆ DATA-ITEM is the 30-character column that you can use to access the view UE-DATA-ITEM. DATA-ITEM contains the following:
 - Characters 1–4: 4-character data set name
 - Characters 5–8: 4-character data item name
 - Characters 10–30: padded with spaces, except for linkpath data items (see “UE-DATA-ITEM” on page 31 for details of key construction for linkpath data items).

UX-DATA-SET-OF-DBDESC

This view connects the views UE-DATABASE and UE-DATA-SET. Use it to provide details of the data items in a specified record. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UX-DATA-SET-OF-DBDESC
FSI: *   VSI: =   MSG:           2544 BYTES USED IN OPENING VIEW.
>GO*

!          DESC                !          DATA-SET          !
!-----!-----!-----!-----!
!TESTDB                !REGN                !
!TESTDB                !BRAN                !
!TESTDB                !ITEM                !
!TESTDB                !CUST                !
!TESTDB                !STCK                !
!TESTDB                !REBR                !
!BOOKDB                !AUTH                !
!BOOKDB                !BOOK                !
!BOOKDB                !REVW                !
!BOOKDB                !BORE                !
!BOOKDB                !AUBO                !
!BOOKDB                !PROD                !
!BOOKDB                !COMS                !
!TRIAL1                !PXXX                !
!-----!-----!-----!-----!
***END***
```

- ◆ DESC is the 30-character key, 6-character database name, padded to the right with spaces to 30 characters.
- ◆ DATA-SET is the 4-character data set name that you use to access the view UE-DATA-SET. This column is padded to the right with spaces to 30 characters.

UX-DERIVED-VIEW-CONSTANT

This view provides any constant value, its column name and name of the base view containing it. This view uses the derived view name as the key. To access a constant value using the key derived view name and the key column name, see “A view that uses two keys to access constant values” on page 108 for a description of the view US-DERIVED-VIEW-CONSTANT. The example DBAID session shows how to open the view and retrieve the records by typing GET*.

```
>OPEN UX-DERIVED-VIEW-CONSTANT
FSI: *   VSI: =   MSG:           2928 BYTES USED IN OPENING VIEW.
>GET*

LV                (+) BRANCH-SUBSET
VALUE             (+) 000004
NAME              (+) BRANCH-REGION
LOGICAL-VIEW      (+) BRANCH
FSI: *   VSI: +   MSG: SUCCESSFUL COMPLETION
```

- ◆ LV is the 30-character key derived view name, padded to the right with spaces to 30 characters.
- ◆ VALUE is a 24-character field and is the constant value.
- ◆ NAME is a 26-character field and contains the column name in the derived view.
- ◆ LOGICAL-VIEW is a 30-character field and is the name of the base view containing the column (the key to the view).

UX-DOMAIN-OF-DATA-ITEM

This view connects the views UE-DATA-ITEM and UE-DOMAIN. Use it to provide details of any domain that is connected to a specified data item. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UX-DOMAIN-OF-DATA-ITEM
FSI: *   VSI: =   MSG:           2544 BYTES USED IN OPENING VIEW.
>GO*

!                DITM                !                DOMAIN                !
!-----!-----!-----!-----!
!REGNCTRL        !REGION-NUMBER        !
!BRANDELV        !DELIVERY-ROUTES     !
!REBRREGN        !REGION-NUMBER        !
!BORECHAP        !CHAPTERS             !
!PRODINTO        !TEST                 !
!CUSTCLAS        !CUSTOMER-CLASS       !
!BORETASK        !REVIEWER-CATEGORIES !
!REBRBRAN        !BRANCH-NUMBERS      !
!STCKBRAN        !STOCK-BRANCHES      !
!BRANCTRL        !BRANCH-NUMBERS      !
!CUSTBRAN        !BRANCH-NUMBERS      !
!BRANREGN        !REGION-NUMBER        !
!-----!-----!-----!-----!
***END***
```

- ◆ DITM is the 30-character key, physical data item name, constructed as follows:
 - Characters 1–4: 4-character data set name
 - Characters 5–8: 4-character data item name
 - Characters 9–30: padded with spaces
- ◆ DOMAIN is the 30-character column that you can use to access the view UE-DOMAIN. This column is padded to the right with spaces if its value is less than 30 characters.

UX-DOMAIN-USES-VALIDATION-TABL

This view connects the views UE-DOMAIN and UE-VALIDATION-TABLE. Use it to provide details of any validation table that is defined for a specified domain. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UX-DOMAIN-USES-VALIDATION-TABL
FSI: * VSI: = MSG:          2544 BYTES USED IN OPENING VIEW.
>GO*

!          DOMN              !          VALIDATION-TABLE      !
!-----!-----!-----!
!TEST                                !TEST                                !
!REVIEWER-CATEGORIES                !JOB-FUNCTIONS                      !
!CHAPTERS                            !PART-OF-MANUAL                     !
!REGION-NUMBER                      !REGIONS                            !
!CUSTOMER-CLASS                     !CUST-CLASSES                       !
!-----!-----!-----!
***END***
```

- ◆ DOMN is the 30-character key, domain name, padded to the right with spaces to 30 characters.
- ◆ VALIDATION-TABLE is the 30-character column that you can use to access the view UE-VALIDATION-TABLE. This column is padded to the right with spaces if its value is less than 30 characters.

UX-FILE-SPEC-OF-DATA-SET

This view connects the views UE-DATA-SET and UE-FILE-SPEC. Use it to access file specification details for a specified data set. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```

>OPEN UX-FILE-SPEC-OF-DATA-SET
FSI: *   VSI: =   MSG:           2544 BYTES USED IN OPENING VIEW.
>GO*

!              DSET              !              FILE-SPEC              !
!-----!-----!-----!-----!
!BRAN          !BRAN TESTDB FILE-SPEC 01      !
!AUTH          !AUTH BOOKDB FILE-SPEC 01      !
!STCK          !STCK TESTDB FILE-SPEC 01      !
!PROD          !PROD BOOKDB FILE-SPEC 01      !
!COMS          !COMS BOOKDB FILE-SPEC 01      !
!ITEM          !ITEM TESTDB FILE-SPEC 01      !
!REGN          !REGN TESTDB FILE-SPEC 01      !
!REBR          !REBR TESTDB FILE-SPEC 01      !
!BORE          !BORE BOOKDB FILE-SPEC 01      !
!REVV          !REVV BOOKDB FILE-SPEC 01      !
!AUBO          !AUBO BOOKDB FILE-SPEC 01      !
!BOOK          !BOOK BOOKDB FILE-SPEC 01      !
!PXXX          !PXXX TRIAL1 FILE-SPEC 01      !
!CUST          !CUST TESTDB FILE-SPEC 01      !
!-----!-----!-----!-----!
***END***

```

- ◆ DSET is the 30-character key, 4-character data set name, padded to the right with spaces to 30 characters.
- ◆ FILE-SPEC is the 30-character column that you can use to access the view UE-FILE-SPEC. FILE-SPEC contains the following:
 - Characters 1–4: 4-character data set name
 - Characters 6–11: 6-character database name
 - Characters 13–21: FILE-SPEC
 - Characters 23–24: 01 for allocation 1
02 for allocation 2
03 for allocation 3
04 for allocation 4
 - Characters 25–30: padded with spaces

UX-FILE-SPEC-OF-DATABASE

This view connects the views UE-DATABASE-DESCRIPTION and UE-FILE-SPEC. Use it to access file specification details for a specified database. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```

>OPEN UX-FILE-SPEC-OF-DATABASE
FSI: * VSI: = MSG:          2544 BYTES USED IN OPENING VIEW.
>GO*

!          DESC          !          FILE-SPEC          !
!-----!-----!-----!-----!
!TESTDB          !REGN TESTDB FILE-SPEC 01          !
!TESTDB          !BRAN TESTDB FILE-SPEC 01          !
!TESTDB          !ITEM TESTDB FILE-SPEC 01          !
!TESTDB          !CUST TESTDB FILE-SPEC 01          !
!TESTDB          !STCK TESTDB FILE-SPEC 01          !
!TESTDB          !REBR TESTDB FILE-SPEC 01          !
!BOOKDB          !AUTH BOOKDB FILE-SPEC 01          !
!BOOKDB          !BOOK BOOKDB FILE-SPEC 01          !
!BOOKDB          !REVW BOOKDB FILE-SPEC 01          !
!BOOKDB          !BORE BOOKDB FILE-SPEC 01          !
!BOOKDB          !AUBO BOOKDB FILE-SPEC 01          !
!BOOKDB          !PROD BOOKDB FILE-SPEC 01          !
!BOOKDB          !COMS BOOKDB FILE-SPEC 01          !
!TRIAL1          !PXXX TRIAL1 FILE-SPEC 01          !
!-----!-----!-----!-----!
***END***
    
```

- ◆ DESC is the 30-character key, 6-character database name, padded to the right with spaces to 30 characters.
- ◆ FILE-SPEC is the 30-character column that you can use to access the view UE-FILE-SPEC. FILE-SPEC contains the following:
 - Characters 1–4: 4-character data set name
 - Characters 6–11: 6-character database name
 - Characters 13–21: FILE-SPEC
 - Characters 23–24: 01 for allocation 1
02 for allocation 2
03 for allocation 3
04 for allocation 4
 - Characters 25–30: padded with spaces

UX-FILE-SPEC-OF-INDEX

This view connects the views UE-INDEX and UE-FILE-SPEC. Use it to access file specification details for a specified index. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UX-FILE-SPEC-OF-INDEX
FSI: *   VSI: =   MSG:           2544 BYTES USED IN OPENING VIEW.
>GO*

!                INDX                !                FILE-SPEC                !
!-----!-----!-----!-----!
!BOOKIXPU                !BOOKIXPU   BOOKDB FILE-SPEC 01!
!PRODIXPP                !PRODIXPP   BOOKDB FILE-SPEC 01!
!-----!-----!-----!-----!
***END***
```

- ◆ INDX is the 30-character key, constructed as follows:
 - Characters 1–4: 4-character data set name
 - Characters 5–6: IX indicating that this is an index
 - Characters 7–8: 2-character index name
 - Characters 9–30: padded with spaces
- ◆ FILE-SPEC is the 30-character column that you can use to access the view UE-FILE-SPEC. FILE-SPEC contains the following:
 - Characters 1–4: 4-character data set name
 - Characters 5–6: IX indicating that this is an index
 - Characters 7–8: 2-character index name
 - Characters 12–17: 6-character database name
 - Characters 19–27: FILE-SPEC
 - Characters 29–30: 01 for allocation 1
02 for allocation 2
03 for allocation 3
04 for allocation 4

UX-INDEX-IN-DATA-SET

This view connects the views UE-DATA-SET and UE-INDEX. Use it to access index details for a specified data set. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UX-INDEX-IN-DATA-SET
FSI: * VSI: = MSG:          2544 BYTES USED IN OPENING VIEW.
>GO*

!           DSET           !           INDEX-FILE           !
!-----!-----!-----!
!PROD           !PRODIXPP           !
!BOOK           !BOOKIXPU           !
!-----!-----!-----!
***END***
```

- ◆ DSET is the 30-character key, 4-character data set name, padded to the right with spaces to 30 characters.
- ◆ INDEX-FILE is the 30-character column that you can use to access the view UE-INDEX. INDEX-FILE contains the following:
 - Characters 1–4: 4-character data set name
 - Characters 5–6: IX indicating that this is an index
 - Characters 7–8: 2-character index name
 - Characters 9–30: padded with spaces

UX-INDEX-OF-DBDESC

This view connects the views UE-DATABASE and UE-INDEX. Use it to access index details for a specified database. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN INDEX-OF-DBDESC
FSI: * VSI: = MSG:          2544 BYTES USED IN OPENING VIEW.
>GO*

!          DESC          !          INDEX-FILE          !
!-----!-----!-----!
!BOOKDB          !PRODIXPP          !
!BOOKDB          !BOOKIXPU          !
!-----!-----!-----!
***END***
```

- ◆ DESC is the 30-character key, 6-character database name, padded to the right with spaces to 30 characters.
- ◆ INDEX-FILE is the 30-character column that you can use to access the view UE-INDEX. INDEX-FILE contains the following:
 - Characters 1–4: 4-character data set name
 - Characters 5–6: IX indicating that this is an index
 - Characters 7–8: 2-character index name
 - Characters 9–30: padded with spaces

UX-LINKPATH-OF-RECORD

This view connects the views UE-RECORD and UE-DATA-ITEM, accessing all the linkpaths in a specified data set. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```

>OPEN UX-LINKPATH-OF-RECORD
FSI: * VSI: = MSG:          2544 BYTES USED IN OPENING VIEW.
>GO*

!          RECD          !          LINKPATH          !
!-----!-----!-----!
!BORE RELATED-DATA      !REVWLKBO RELATED-DATA      !
!BORE RELATED-DATA      !BOOKLKRE RELATED-DATA      !
!COMS RELATED-DATA      !BOOKLKCO RELATED-DATA      !
!REGN PRIMARY-DATA      !REGNLKBR PRIMARY-DATA      !
!ITEM PRIMARY-DATA      !ITEMLKST PRIMARY-DATA      !
!REVV PRIMARY-DATA      !REVWLKBO PRIMARY-DATA      !
!REBR RELATED-DATA      !REGNLKBR RELATED-DATA      !
!REBR RELATED-DATA      !BRANLKRE RELATED-DATA      !
!AUTH PRIMARY-DATA      !AUTHLKBO PRIMARY-DATA      !
!PROD RELATED-DATA      !BOOKLKPR RELATED-DATA      !
!STCK RELATED-DATA      !BRANLKST RELATED-DATA      !
!STCK RELATED-DATA      !ITEMLKST RELATED-DATA      !
!BRAN PRIMARY-DATA      !BRANLKRE PRIMARY-DATA      !
!BRAN PRIMARY-DATA      !BRANLKST PRIMARY-DATA      !
!BOOK PRIMARY-DATA      !BOOKLKAU PRIMARY-DATA      !
!BOOK PRIMARY-DATA      !BOOKLKPR PRIMARY-DATA      !
!BOOK PRIMARY-DATA      !BOOKLKCO PRIMARY-DATA      !
!-----!-----!-----!
***MORE***
    
```

- ◆ RECD is the 30-character key, constructed as follows:
 - Characters 1–4: 4-character data set name
 - Characters 6–17: PRIMARY-DATA for a primary data set
RELATED-DATA for the base portion of a related data set
RMS-DATA (plus 4 spaces) for an RMS data set
 - Characters 18–30: padded with spaces

If RECD is a coded record, its key construction is:

- Characters 1–4: 4-character data set name
 - Characters 6–17: CODED-DATA
 - Characters 17–18: 2-character record code
 - Characters 19–30: padded with spaces
- ◆ LINKPATH is the 30-character column that you can use to access the view UE-DATA-ITEM. LINKPATH contains the following:
 - Characters 1–4: 4-character data set name
 - Characters 5–8: 4-character data item name
 - Characters 10–21: PRIMARY-DATA for a primary data set
RELATED-DATA for a linkpath in the base portion of a related data set
 - Characters 22–30: padded with spaces

If LINKPATH occurs in the coded portion of a related data set, its key construction is:

- Characters 1–4: 4-character data set name
- Characters 5–8: 4-character data item name
- Characters 10–19: CODED-DATA
- Characters 21–22: 2-character record code
- Characters 23–30: padded with spaces

UX-LOGICAL-DATA-ITEM-OF-DATA-S

This view connects the views UE-DATA-SET and UE-LOGICAL-DATA-ITEM. Use it to access logical data item details for a specified data set. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```

>OPEN UX-LOGICAL-DATA-ITEM-OF-DATA-S
FSI: *   VSI: =   MSG:           2544 BYTES USED IN OPENING VIEW.
>GO*

!           DSET                !           DATA-ITEM                !
!-----!-----!-----!-----!
!BRAN      !BRAN      !BRANCH-NO      !
!BRAN      !BRAN      !BRANCH-NAME    !
!BRAN      !BRAN      !BRANCH-ADDR    !
!BRAN      !BRAN      !BRANCH-CITY    !
!BRAN      !BRAN      !BRANCH-STATE   !
!BRAN      !BRAN      !BRANCH-ZIPCODE !
!BRAN      !BRAN      !BRANCH-REGION  !
!BRAN      !BRAN      !BRANCH-DEL-ROUTE !
!BRAN      !BRAN      !BRANCH-SLS-QUOTA !
!BRAN      !BRAN      !BRANCH-STF-QUOTA !
!AUTH      !AUTH      !AUTHOR         !
!STCK      !STCK      !STOCK-PRODUCT  !
!STCK      !STCK      !STOCK-BRANCH   !
!STCK      !STCK      !STOCK-BIN-LOC  !
!STCK      !STCK      !STOCK-YTD-SALES !
!PROD      !PROD      !INTO-PRODUCTION !
!PROD      !PROD      !OUT-OF-PRODUCTION !
!-----!-----!-----!-----!
***MORE***
    
```

- ◆ DSET is the 30-character key, 4-character data set name, padded to the right with spaces to 30 characters.
- ◆ DATA-ITEM is the 30-character column that you can use to access the view UE-LOGICAL-DATA-ITEM. This column is padded to the right with spaces if its value is less than 30 characters.

UX-LOGICAL-DATA-ITEM-OF-LV

This view connects the views UE-LOGICAL-VIEW and UE-LOGICAL-DATA-ITEM. Use it to access logical data item details for a specified base or derived view. The example DBAID session shows how to open the view and retrieve the records by typing GET*.

```
>OPEN UX-LOGICAL-DATA-ITEM-OF-LV
FSI: *   VSI: =   MSG:           3120 BYTES USED IN OPENING VIEW.
>GET*

LV                (+) REGION
ALIAS              (+)
TYPE               (+) 1
VALUE             (+)
DATA-ITEM          (+) REGION-NO
FSI: *   VSI: +   MSG: SUCCESSFUL COMPLETION
```

- ◆ LV is the 30-character key view name, padded to the right with spaces to 30 characters.
- ◆ ALIAS, a 30-character field, is the column name in the view if this name differs from the logical data item name. This field remains blank if the column name matches the logical data item name.

- ◆ TYPE, a 2-byte integer field, indicates the logical data item type and uses these codes to identify logical data item type
 - 0 Data
 - 1 Key
 - 2 Required
 - 3 Constant
 - 4 Nonunique key
 - 5 Unique constant
 - 6 Foreign key
 - 256 Data
 - 257 Key
 - 258 Required
 - 259 Constant
 - 260 Nonunique key
 - 261 Unique constant
 - 262 Foreign key
- ◆ VALUE is a 24-character field.
- ◆ DATA-ITEM is the name of the logical data item and is the 30-character column that you can use to access the view UE-LOGICAL-DATA-ITEM. This column is padded to the right with spaces if its value is less than 30 characters.

UX-LOGICAL-VIEW-OF-DBDESC

This view connects the views UE-DATABASE-DESCRIPTION and UE-LOGICAL-VIEW. Use it to access the views connected to a specified database. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UX-LOGICAL-VIEW-OF-DBDESC
FSI: *   VSI: =   MSG:           2544 BYTES USED IN OPENING VIEW.
>GO*

!          DESC                      !          LOGICAL-VIEW          !
!-----!-----!-----!-----!
!TESTDB   !BASE-VIEW-B                       !
!TESTDB   !BASE-VIEW-A                       !
!TESTDB   !INSERT-BRANCH                     !
!TESTDB   !REGION-SUBSET                     !
!TESTDB   !INDEXED-BRANCHES-IN-REGION       !
!TESTDB   !INDEXED-PRODUCTS-IN-REGION      !
!TESTDB   !INDEXED-BRANCH-SUBSET         !
!TESTDB   !INDEXED-CUSTOMER              !
!TESTDB   !INDEXED-PRODUCT               !
!TESTDB   !INDEXED-BRANCH               !
!TESTDB   !INDEXED-STOCK                !
!TESTDB   !INDEXED-REGION               !
!TESTDB   !BASE-VIEW                      !
!TESTDB   !REGION                      !
!TESTDB   !BRANCH                      !
!TESTDB   !STOCK                      !
!TESTDB   !CUSTOMER                    !
!-----!-----!-----!-----!
***MORE***
```

- ◆ DESC is the 30-character key, 6-character database name, padded to the right with spaces to 30 characters.
- ◆ LOGICAL-VIEW is the 30-character column that you can use to access the view UE-LOGICAL-VIEW. This column is padded to the right with spaces if its value is less than 30 characters.

UX-PHYSICAL-TO-LOGICAL-DATA-IT

This view connects the views UE-DATA-ITEM and UE-LOGICAL-DATA-ITEM. Use it to access logical data item name equivalents for a specified physical data item. To access the physical data items from the logical data item names (perform a reverse scan on the linkpath), see “[UR-PHYSICAL-LOGICAL-DATA-ITEM](#)” on page 105 for a description of the view UR-PHYSICAL-LOGICAL-DATA-ITEM. The example DBAID session shows how to open the view UX-PHYSICAL-TO-LOGICAL-DATA-IT and retrieve the records by typing GO*.

```

>OPEN UX-PHYSICAL-TO-LOGICAL-DATA-IT
FSI: * VSI: = MSG:          2544 BYTES USED IN OPENING VIEW.
>GO*

!           DITM           !           DATA-ITEM           !
!-----!-----!-----!
!AUBOAUTH           !AUTHOR-OF-BOOK           !
!CUSTCHRI           !CUSTOMER-CHRISTIAN-NAME  !
!BOREBOOK           !BOOK-REVIEWED           !
!COMSCOM1           !COMMENT-1                 !
!REVWNAME           !CHRISTIAN-NAME           !
!AUBOBOOK           !BOOK-WRITTEN             !
!BOREREVW           !REVIEWER-OF-BOOK        !
!AUBOADAT           !FINISH-WRITING           !
!ITEMCTRL           !PRODUCT-CODE             !
!BOOKPUBL           !PUB-NO                    !
!BRANSLSQ           !BRANCH-SLS-QUOTA         !
!BRANSTFQ           !BRANCH-STF-QUOTA         !
!PRODOUTO           !OUT-OF-PRODUCTION        !
!AUBOSDAT           !START-WRITING            !
!BRANNAME           !BRANCH-NAME              !
!REGNCTRL           !REGION-NO                 !
!BRANDELV           !BRANCH-DEL-ROUTE         !
!-----!-----!-----!
***MORE***
    
```

- ◆ DITM is the 30-character key, 4-character physical data item name, constructed as follows:
 - Characters 1–4: 4-character data set name
 - Characters 5–8: 4-character data item name
 - Characters 9–30: padded with spaces

- ◆ DATA-ITEM is the 30-character column that you can use to access the view UE-LOGICAL-DATA-ITEM. This column is padded to the right with spaces if its value is less than 30 characters.

UX-PROGRAM-USES-LOGICAL-VIEW

This view connects the views UE-PROGRAM and UE-LOGICAL-VIEW. Use it to list the views used by a specified program. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UX-PROGRAM-USES-LOGICAL-VIEW
FSI: *   VSI: =   MSG:           2544 BYTES USED IN OPENING VIEW.
>GO*

!           PROG                !           LOGICAL-VIEW           !
!-----!-----!-----!-----!
!GET-MANUALS                !MANUALS-VIEW                !
!MANUALS                    !MANUALS                      !
!-----!-----!-----!-----!
***END***
```

- ◆ PROG is the 30-character key, program name, padded to the right with spaces when the key value is less than 30 characters.
- ◆ LOGICAL-VIEW is the 30-character column that you can use to access the view UE-LOGICAL-VIEW. This column is padded to the right with spaces if its value is less than 30 characters.

UX-RECORD-OF-DATA-SET

This view connects the views UE-DATA-SET and UE-RECORD. Use it to access record details for a specified data set. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```

>OPEN UX-RECORD-OF-DATA-SET
FSI: * VSI: = MSG:          2544 BYTES USED IN OPENING VIEW.
>GO*

!           DSET           !           RECORD           !
!-----!-----!-----!-----!
!BRAN          !BRAN PRIMARY-DATA      !
!AUTH          !AUTH PRIMARY-DATA      !
!STCK          !STCK RELATED-DATA      !
!PROD          !PROD RELATED-DATA      !
!COMS          !COMS RELATED-DATA      !
!ITEM          !ITEM PRIMARY-DATA      !
!REGN          !REGN PRIMARY-DATA      !
!REBR          !REBR RELATED-DATA      !
!BORE          !BORE RELATED-DATA      !
!REVV          !REVV PRIMARY-DATA      !
!AUBO          !AUBO RELATED-DATA      !
!BOOK          !BOOK PRIMARY-DATA      !
!CUST          !CUST RMS-DATA          !
!-----!-----!-----!-----!
***END***
    
```

- ◆ DSET is the 30-character key, 4-character data set name, padded to the right with spaces to 30 characters.
- ◆ RECORD is the 30-character column that you can use to access the view UE-RECORD. RECORD contains the following:
 - Characters 1–4: 4-character data set name
 - Characters 6–17: PRIMARY-DATA for a primary data set
RELATED-DATA for the base portion of a related data set
RMS-DATA (4 spaces) for an RMS data set
 - Characters 18–30: padded with spaces

If RECORD is a coded record, its key construction is:

- Characters 1–4: 4-character data set name
- Characters 6–17: CODED-DATA
- Characters 17–18: 2-character record code
- Characters 19–30: padded with spaces

UX-RMS-KEY-IN-DATA-ITEM

This view connects the views UE-DATA-ITEM and UE-RMS-KEY. Use it to access details of any RMS keys that are defined with a specified data item. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UX-RMS-KEY-IN-DATA-ITEM
FSI: *   VSI: =   MSG:           2544 BYTES USED IN OPENING VIEW.
>GO*

!           DITM                !           RMS-KEY                !
!-----!-----!-----!-----!
!CUSTNAME                !CUSTNAMEKEY                !
!CUSTCTRL                !CUSTBASEKEY                !
!-----!-----!-----!-----!
***END***
```

- ◆ DITM is the 30-character key, 4-character physical data item name, constructed as follows:
 - Characters 1–4: 4-character data set name
 - Characters 5–8: 4-character data item name
 - Characters 9–30: padded with spaces
- ◆ RMS-KEY is the 30-character column that you can use to access the view UE-RMS-KEY. RMS-KEY contains the following:
 - Characters 1–4: 4-character data set name
 - Characters 5–12: RMS key name as specified through DBA
 - Characters 13–30: padded with spaces

UX-RMS-KEY-IN-DATA-SET

This view connects the views UE-DATA-SET and UE-RMS-KEY. Use it to access details of any RMS keys that are defined for a specified data set. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UX-RMS-KEY-IN-DATA-SET
FSI: * VSI: = MSG:          2544 BYTES USED IN OPENING VIEW.
>GO*

!           DSET           !           RMS-KEY           !
!-----!-----!-----!
!CUST           !CUSTBASEKEY           !
!CUST           !CUSTNAMEKEY           !
!-----!-----!-----!
***END***
```

- ◆ DSET is the 30-character key, 4-character data set name, padded to the right with spaces to 30 characters.
- ◆ RMS-KEY is the 30-character column that you can use to access the view UE-RMS-KEY. RMS-KEY contains the following:
 - Characters 1–4: 4-character data set name
 - Characters 5–12: RMS key name as specified through DBA
 - Characters 13–30: padded with spaces

UX-SEC-KEY-IN-INDEX

This view connects the views UE-INDEX and UE-SECONDARY-KEY. Use it to access secondary key details for a specified data set. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UX-SEC-KEY-IN-INDEX
FSI: *   VSI: =   MSG:           2544 BYTES USED IN OPENING VIEW.
>GO*

!                INDX                !                SECONDARY-KEY                !
!-----!-----!-----!-----!
!BOOKIXPU                !BOOKSKNO                !
!PRODIXPP                !PRODSKNO                !
!-----!-----!-----!-----!
***END***
```

- ◆ INDX is the 30-character key, constructed as follows:
 - Characters 1–4: 4-character data set name
 - Characters 5–6: IX indicating that this is an index
 - Characters 7–8: 2-character index name
 - Characters 9–30: padded with spaces
- ◆ SECONDARY-KEY is the 30-character column that you can use to access the view UE-SECONDARY-KEY. SECONDARY-KEY contains the following:
 - Characters 1–4: 4-character data set name
 - Characters 5–6: SK indicating that this is a secondary key
 - Characters 7–8: 2-character key name
 - Characters 9–30: padded with spaces

UX-SUB-DATA-ITEM

This view shows you which data items are subdefined. You can use the SUB-DATA-ITEM column to access the view UE-DATA-ITEM and display sub-data-item details. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```

>OPEN UX-SUB-DATA-ITEM
FSI: * VSI: = MSG:          2544 BYTES USED IN OPENING VIEW.
>GO*

!           DITM           !           SUB-DATA-ITEM           !
!-----!-----!-----!
!CUSTNAME           !CUSTSURN           !
!CUSTNAME           !CUSTCHRI           !
!ITEMPRCE           !ITEMDDDD           !
!ITEMSUBD           !ITEMONCE           !
!ITEMSUBD           !ITEMTWIC           !
!-----!-----!-----!
***END***
    
```

- ◆ DITM is the 30-character key, data item name, constructed as follows:
 - Characters 1–4: 4-character data set name
 - Characters 5–8: 4-character data item name
 - Characters 10–30: padded with spaces
- ◆ SUB-DATA-ITEM is the 30-character column that you can use to access the view UE-DATA-ITEM, and contains the following:
 - Characters 1–4: 4-character data set name
 - Characters 5–8: 4-character data item name
 - Characters 9–30: padded with spaces

UX-SYSTEM-LOG-OF-DBDESC

This view connects the views UE-DATABASE-DESCRIPTION and UE-SYSTEM-LOG. Use it to access system log details for a specified database. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UX-SYSTEM-LOG-OF-DBDESC
FSI: *   VSI: =   MSG:           2544 BYTES USED IN OPENING VIEW.
>GO*

!           DESC                !           SYSTEM-LOG           !
!-----!-----!-----!-----!
!TESTDB                !TESTDB SYSTEM-LOG             !
!BOOKDB                !BOOKDB FUNCTION-LOG          !
!-----!-----!-----!-----!
***END***
```

- ◆ DESC is the 30-character key, 6-character database name, padded to the right with spaces to 30 characters.
- ◆ SYSTEM-LOG is the 30-character column that you can use to access the view UE-SYSTEM-LOG. This column contains the following:
 - Characters 1–6: 6-character database name
 - Characters 8–19: FUNCTION-LOG if the database was defined using ULTRA and upgraded to SUPRA or SYSTEM-LOG if the database was defined using SUPRA
 - Characters 20–30: padded with spaces

UX-TASK-LOG-OF-DBDESC

This view connects the views UE-DATABASE-DESCRIPTION and UE-TASK-LOG. Use it to access system log details for a specified database. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UX-TASK-LOG-OF-DBDESC
FSI: * VSI: = MSG:          2544 BYTES USED IN OPENING VIEW.
>GO*

!           DESC           !           TASK-LOG           !
!-----!-----!-----!
!TESTDB           !TESTDB TASK-LOG           !
!SPARED           !SPARED TASK-LOG           !
!BOOKDB           !BOOKDB TASK-LOG           !
!TRIAL1           !TRIAL1 TASK-LOG           !
!-----!-----!-----!
***END***
```

- ◆ DESC is the 30-character key, 6-character database name, padded to the right with spaces to 30 characters.
- ◆ TASK-LOG is the 30-character column that you can use to access the view UE-TASK-LOG. This column contains the following:
 - Characters 1–6: 6-character database name
 - Characters 8–15: TASK-LOG
 - Characters 16–30: padded with spaces

UX-USER-MAY-RUN-PROGRAM

This view connects the views UE-USER and UE-PROGRAM. Use it to access details of the programs defined on the Directory, available to user PUBLIC. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UX-USER-MAY-RUN-PROGRAM
FSI: * VSI: = MSG:          2544 BYTES USED IN OPENING VIEW.
>GO*
!                USER                !                PROGRAM                !
!-----!-----!-----!-----!
!PUBLIC                !MANUALS                !
!PUBLIC                !GET-MANUALS            !
!-----!-----!-----!-----!
***END***
```

- ◆ USER is the 30-character key, user name, padded to the right with spaces when the key value is less than 30 characters.
- ◆ PROGRAM is the 30-character column that you can use to access the view UE-PROGRAM. This column is padded to the right with spaces if its value is less than 30 characters.

UX-USER-MAY-USE-LOGICAL-VIEW

This view connects the views UE-USER and UE-LOGICAL-VIEW. Use it to display details of the views available to a specified user. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```

>OPEN UX-USER-MAY-USE-LOGICAL-VIEW
FSI: * VSI: = MSG:          2544 BYTES USED IN OPENING VIEW.
>GO*

!          USER          !          LOGICAL-VIEW          !
!-----!-----!-----!
!FRED          !REGION          !
!FRED          !BRANCH          !
!FRED          !PRODUCTS-IN-REGION          !
!FRED          !INDEXED-REGION          !
!FRED          !INDEXED-BRANCH          !
!FRED          !INDEXED-BRANCHES-IN-REGION          !
!FRED          !INDEXED-BRANCH-SUBSET          !
!CLOUGH          !EDITS          !
!CLOUGH          !BASE-VIEW-B          !
!CLOUGH          !BASE-VIEW-A          !
!CLOUGH          !INSERT-BRANCH          !
!CLOUGH          !REGION-SUBSET          !
!CLOUGH          !INDEXED-BRANCHES-IN-REGION          !
!CLOUGH          !INDEXED-PRODUCTS-IN-REGION          !
!CLOUGH          !INDEXED-BRANCH-SUBSET          !
!CLOUGH          !INDEXED-CUSTOMER          !
!CLOUGH          !INDEXED-PRODUCT          !
!-----!-----!-----!
***MORE***
    
```

- ◆ USER is the 30-character key, user name, padded to the right with spaces when the key value is less than 30 characters.
- ◆ LOGICAL-VIEW is the 30-character column that you can use to access the view UE-LOGICAL-VIEW. This column is padded to the right with spaces if its value is less than 30 characters.

UX-VIEW-TO-VIEW

This view accesses the base views used as data sources for columns included in derived views. To access the base views from the derived views using a different linkpath, see “[UR-VIEW-TO-VIEW](#)” on page 106 for a description of the view UR-VIEW-TO-VIEW. The example DBAID session shows how to open the view UX-VIEW-TO-VIEW and retrieve the records by typing GET*.

```

>OPEN UX-VIEW-TO-VIEW
FSI: *   VSI: =   MSG:           3120 BYTES USED IN OPENING VIEW.
>GET*

LV                (+) PRINTED-MANUALS
NAME              (+)
SOURCE-NAME       (+) MANUAL-TITLE
TYPE              (+) 1
LOGICAL-VIEW-D    (+) PRODUCTION-DETAILS
FSI: *   VSI: +   MSG: SUCCESSFUL COMPLETION
>GET*

LV                (=) PRINTED-MANUALS
NAME              (+)
SOURCE-NAME       (+) TO-PRINT
TYPE              (+) 0
LOGICAL-VIEW-D    (+) PRODUCTION-DETAILS
FSI: *   VSI: +   MSG: SUCCESSFUL COMPLETION

```

- ◆ LV is the 30-character key derived view name, padded to the right with spaces when the key value is less than 30 characters.
- ◆ NAME is a 26-character field and contains any alternative column name if one is specified in the derived view.
- ◆ SOURCE-NAME is a 26-character field and is the name of the base view column.

- ◆ TYPE, a 2-byte integer field, is the data type of the column and uses these codes to identify logical data item type:
 - 0 Data
 - 1 Key
 - 2 Required
 - 3 Constant
 - 4 Nonunique key
 - 5 Unique constant
 - 6 Foreign key
 - 256 Data
 - 257 Key
 - 258 Required
 - 259 Constant
 - 260 Nonunique key
 - 261 Unique constant
 - 262 Foreign key
- ◆ LOGICAL-VIEW-D is a 30-character field and is the name of the base view containing the column (the key column).

4

Special views

This chapter describes views that fall into neither the Entity category nor the Relationship category and includes:

- ◆ Views that access comments, prefixed UC.
- ◆ A view that provides access definitions, prefixed UN.
- ◆ Views that show alternative relationships between entities using a different linkpath than their corresponding UX views, prefixed UR.
- ◆ A view that uses two keys to access constant key values, prefixed US.

Views that access comments

The names of Directory Views that access comments start with the code UC. They access comment lines defined through the comment screen in DBA. This section describes the comment view UC-DATABASE-DESCRIPTION, which accesses the comments for the entity database description. In addition to the comment view UC-DATABASE-DESCRIPTION, you can use the following views to access comments:

- ◆ UC-BUFFER
- ◆ UC-DATA-ITEM
- ◆ UC-DATA-SET
- ◆ UC-DOMAIN
- ◆ UC-FILE-SPEC
- ◆ UC-INDEX
- ◆ UC-INDEX-FILE-SPEC
- ◆ UC-LOGICAL-DATA-ITEM
- ◆ UC-LOGICAL-VIEW
- ◆ UC-PROGRAM
- ◆ UC-RMS-DATA-SET
- ◆ UC-RMS-FILE-SPEC
- ◆ UC-RMS-KEY
- ◆ UC-SECONDARY-KEY
- ◆ UC-SYSTEM-LOG
- ◆ UC-TASK-LOG
- ◆ UC-USER
- ◆ UC-VALIDATION-TABLE

For data type descriptions of the columns of these comment views, see [“Data type descriptions”](#) on page 109.

The example DBAID session shows how to open UC-DATABASE-DESCRIPTION and retrieve comment records for a database. Substitute any other UC view name for UC-DATABASE-DESCRIPTION to retrieve comments for other Directory entities.

```

>OPEN UC-DATABASE-DESCRIPTION
FSI: * VSI: = MSG:          2408 BYTES USED IN OPENING VIEW.
>GO*

!           DESC           !
!-----!
!TESTDB           !
=====

!                               LINE                               !
!-----!
!This database is based on the RDM Service Level 5 database used in !
!the SUPRA RDM Admin Guide, P26-8220-03. It consists of four primary !
!data sets and one RMS data set. There are no related data sets, !
!therefore all navigation uses the WHERE clause. The base views and !
!derived views are also drawn from the SUPRA RDM manual. !
!-----!
***MORE***

```

- ◆ Comment (UC-) views use a 30-character key, padded to the right with spaces when the key value is less than 30 characters.
- ◆ DESC is a 30-character key field.
- ◆ LINE is a 72-character field.

A view that provides access definitions

The view UN-LOGICAL-VIEW obtains the access definitions for all views defined on the Directory. The example DBAID session shows how to open the view and retrieve an access definition by typing GO*.

```

>OPEN UN-LOGICAL-VIEW
FSI: * VSI: = MSG:          2408 BYTES USED IN OPENING VIEW.
>GO*

!                LV                !
!-----!
!REGION                !
=====
!                                LINE                                !
!-----!
!ACCESS REGN USING REGION-NO ALLOW ALL                !
!ACCESS REBR VIA REGNLKBR                !
!* To restrict deletions of REGIONs that contain branches:                !
!ACCESS BRAN USING BRANCH-TO-REGION                !
-----
***MORE***

```

- ◆ LV is the 30-character key, view name (may be base view name or derived view name), padded to the right with spaces when the key value is less than 30 characters.
- ◆ LINE is a 72-character field.

Views that show alternative relationship entities

The following views show relationship entities using a different linkpath than their corresponding views with the prefix UX:

- ◆ UR-PHYSICAL-LOGICAL-DATA-ITEM, which shows the physical data item for a logical name. Also see “[UX-PHYSICAL-TO-LOGICAL-DATA-IT](#)” on page 88.
- ◆ UR-VIEW-TO-VIEW, which shows column relationships between base and derived views. Also see the “[UX-VIEW-TO-VIEW](#)” on page 99.

UR-PHYSICAL-LOGICAL-DATA-ITEM

UR-PHYSICAL-LOGICAL-DATA-ITEM shows the connections between physical and logical data items. It uses the logical data item name as the key to obtain the physical data item equivalent. The example DBAID session shows how to open the view and retrieve the records by typing GO*.

```
>OPEN UR-PHYSICAL-LOGICAL-DATA-ITEM
FSI: * VSI: = MSG:          1920 BYTES USED IN OPENING VIEW.
>GO*

!          LGDI-NAME          !          UDD2-NMPM          !
!-----!-----!-----!
!COMMENT-1          !COMSCOM1          !
!BRANCH-SLS-QUOTA  !BRANSLSQ          !
!CUSTOMER-ZIPCODE  !CUSTZIPC          !
!START-WRITING     !AUBOSDAT          !
!CUSTOMER-SURNAME  !CUSTSURN          !
!CUSTOMER-FIRST-NAME !CUSTCNAM          !
!CUSTOMER-CR-LIM   !CUSTCLIM          !
!REGION-TO-BRANCH  !REBRREGN          !
!COMMENT-9         !COMSCOM9          !
!MANUAL-IN-PRODUCTION !PRODBOOK          !
!REVIEW-SUBMIT     !BORESDAT          !
!FIRST-NAME        !REVWNAME          !
!BRANCH-ADDR       !BRANADDR          !
!PRODUCT-GROUP     !ITEMGRUP          !
!BRANCH-NO         !BRANCTRL          !
!CUSTOMER-NO       !CUSTCTRL          !
!MANUAL            !BOOKCTRL          !
!-----!-----!-----!
***MORE***
```

- ◆ LGDI-NAME is the 30-character key logical data item name, padded to the right with spaces when the key value is less than 30 characters.
- ◆ UDD2-NMPM is the physical data item name for the logical data item name, and is constructed as follows:
 - Characters 1–4: 4-character data set name
 - Characters 5–8: 4-character data item name
 - Characters 9–30: padded with spaces

UR-VIEW-TO-VIEW

UR-VIEW-TO-VIEW provides where-used details. It uses the base view name as the key to determine which derived views use the base view as a data source. The example DBAID session shows how to open the view and retrieve the records by typing GET*.

```
>OPEN UR-VIEW-TO-VIEW
FSI: * VSI: = MSG:          3120 BYTES USED IN OPENING VIEW.
>GET*
LV                (+) INDEXED-PRODUCTS-IN-REGION
NAME              (+)
SOURCE-NAME       (+) REGION-NO
TYPE              (+) 1
LOGICAL-VIEW-D    (+) REGION
FSI: * VSI: + MSG: SUCCESSFUL COMPLETION
```

- ◆ LV is a 30-character field and is the derived view name.
- ◆ NAME is a 26-character field and contains any alternative column name if one is specified in the derived view.
- ◆ SOURCE-NAME is a 26-character field and is the name of the base view column.

- ◆ TYPE is a 2-byte integer field and is the data type of the column and uses these codes to identify logical data item type:
 - 0 Data
 - 1 Key
 - 2 Required
 - 3 Constant
 - 4 Nonunique key
 - 5 Unique constant
 - 6 Foreign key
 - 256 Data
 - 257 Key
 - 258 Required
 - 259 Constant
 - 260 Nonunique key
 - 261 Unique constant
 - 262 Foreign key
- ◆ LOGICAL-VIEW-D is the name of the base view containing the column (key to the view). It is the 30-character key, base view name, padded to the right with spaces when the key value is less than 30 characters.

A view that uses two keys to access constant values

US-DERIVED-VIEW-CONSTANT uses the derived view name and the column name to provide the constant value and the name of the base view containing it. The example DBAID session shows how to open the view and retrieve the records by typing GET*.

```
>OPEN US-DERIVED-VIEW-CONSTANT
FSI: * VSI: = MSG:          2928 BYTES USED IN OPENING VIEW.
>GET*

LV                (+) BRANCH-SUBSET
VALUE             (+) 000004
NAME              (+) BRANCH-REGION
LOGICAL-VIEW-D   (+) BRANCH
FSI: * VSI: + MSG: SUCCESSFUL COMPLETION
```

- ◆ LV is the 30-character key, derived view name (key to the view), padded to the right with spaces when the key value is less than 30 characters.
- ◆ VALUE is a 24-character field and is the constant value.
- ◆ NAME is the other 30-character key, logical data item name, padded to the right with spaces when the key value is less than 30 characters. NAME contains any alternative column name if one is specified in the derived view.
- ◆ LOGICAL-VIEW-D is a 30-character field and is the name of the base view containing the column (key to the view).

A

Data type descriptions

This Appendix provides data type descriptions of the view columns for the COBOL, FORTRAN, and BASIC programming languages. The table in this appendix presents the data type descriptions.

The table is sorted alphabetically by the logical name of each view. It provides the data types in the programming languages for each column of each directory view.

View name	Column name	COBOL data type	FORTRAN data type	BASIC data type
UC-BUFFER	BUFF	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
UC-DATA-ITEM	DITM	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
UC-DATA-SET	DSET	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
UC-DATABASE-DESCRIPTION	DESC	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
UC-DOMAIN	DOMN	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
UC-FILE-SPEC	FSPC	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
UC-INDEX	INDX	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
UC-INDEX-FILE-SPEC	FSPC	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
	LINE	PIC X(72)	Character*72	Stringx=72

View name	Column name	COBOL data type	FORTRAN data type	BASIC data type
UC-LOGICAL-DATA-ITEM	LGDI	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
UC-LOGICAL-VIEW	LV	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
UC-PROGRAM	PROG	PIC X(30)	Character*30	Stringx=30
UC-RMS-DATA-SET	DSET	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
UC-RMS-FILE-SPEC	FSPC	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
UC-RMS-KEY	RMSK	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
UC-SECONDARY-KEY	SKEY	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
UC-SYSTEM-LOG	FLOG	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
UC-TASK-LOG	TLOG	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
UC-USER	USER	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
UC-VALIDATION-TABLE	VTBL	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
UE-BUFFER	BUFF	PIC X(30)	Character*30	Stringx=30
	COPIES	PIC S9(4) Comp	Integer*2	Word
	TYPE	PIC S9(4) Comp	Integer*2	Word
	SIZE	PIC S9(4) Comp	Integer*2	Word

View name	Column name	COBOL data type	FORTTRAN data type	BASIC data type	
UE-DATA-ITEM	DITM	PIC X(30)	Character*30	Stringx=30	
	LENGTH	PIC S9(4) Comp	Integer*2	Word	
	USE	PIC S9(4)	Integer*2	Word	
	TYPE	PIC X(1)	Character*1	Stringx=1	
	SIGN	PIC X(1)	Character*1	Stringx=1	
	DECIMAL-PLACES	PIC S9(4)	Integer*2	Word	
	RELATED-KEY	PIC X(8)	Character*8	Stringx=8	
	DISPLACEMENT	PIC S9(4) Comp	Integer*2	Word	
	STAR-FILLER	PIC X(1)	Character*1	Stringx=1	
	SUB-ITEMS	PIC S9(4) Comp	Integer*2	Word	
	DATA-ITEM-LEVEL	PIC S9(4) Comp	Integer*2	Word	
	UE-DATA-SET	DSET	PIC X(30)	Character*30	Stringx=30
		LENGTH	PIC S9(4) Comp	Integer*2	Word
TYPE		PIC S9(4) Comp	Integer*2	Word	
CODED		PIC X(1)	Character*1	Stringx=1	
STATUS		PIC X(1)	Character*1	Stringx=1	
KEY-LENGTH		PIC S9(4) Comp	Integer*2	Word	

View name	Column name	COBOL data type	FORTRAN data type	BASIC data type
UE-DATABASE-DESCRIPTION	DESC	PIC X(30)	Character*30	Stringx=30
	PASSWORD	PIC X(6)	Character*6	Stringx=6
	MAX-HELD-RECORDS	PIC S9(4) Comp	Integer*2	Word
	MAX-TASKS	PIC S9(4) Comp	Integer*2	Word
	MAX-UPDATE-TASKS	PIC S9(4) Comp	Integer*2	Word
	SHADOW-OPTION	PIC X(1)	Character*1	Stringx=1
	SINGLE-TASK	PIC X(1)	Character*1	Stringx=1
	STATUS	PIC X(1)	Character*1	Stringx=1
	DATE-COMPILED	PIC S9(9) Comp	Integer*4	Long
	TIME-COMPILED	PIC S9(9) Comp	Integer*4	Long
	ACCESS-METHOD	PIC X(1)	Character*1	Stringx=1
	GLOBAL-SECT-TYPE	PIC X(1)	Character*1	Stringx=1
	CALLING-MECH	PIC X(1)	Character*1	Stringx=1
	CLUSTER-NETWORK	PIC X(1)	Character*1	Stringx=1

View name	Column name	COBOL data type	FORTTRAN data type	BASIC data type
UE-DOMAIN	DOMN	PIC X(30)	Character*30	Stringx=30
	FUNCTION	PIC S9(4) Comp	Integer*2	Word
	UNIT	PIC S9(4) Comp	Integer*2	Word
	FORMAT	PIC X(1)	Character*1	Stringx=1
	LENGTH	PIC S9(9) Comp	Integer*4	Long
	DEC-PLACES	PIC S9(9) Comp	Integer*4	Long
	SIGN-OPT	PIC X(1)	Character*1	Stringx=1
	NULLS-ALLOWED	PIC X(1)	Character*1	Stringx=1
	NULL-VALUE	PIC X(32)	Character*32	Stringx=32
	DEFAULT	PIC X(32)	Character*32	Stringx=32
	GET-VAL-OPT	PIC X(1)	Character*1	Stringx=1
	VAL-OPT	PIC X(1)	Character*1	Stringx=1
	VAL-MIN	PIC X(32)	Character*32	Stringx=32
	VAL-MAX	PIC X(32)	Character*32	Stringx=32
	VAL-EXIT	PIC X(8)	Character*8	Stringx=8
	STATUS	PIC X(1)	Character*1	Stringx=1

View name	Column name	COBOL data type	FORTRAN data type	BASIC data type
UE-FILE-SPEC	FSPC	PIC X(30)	Character*30	Stringx=30
	TOTAL-RECORDS	PIC S9(9) Comp	Integer*4	Long
	RECS-PER-BLOCK	PIC S9(4) Comp	Integer*2	Word
	CONTROL-INTERVAL	PIC S9(9) Comp	Integer*4	Long
	LOAD-LIMIT	PIC S9(4) Comp	Integer*2	Word
	ALLOCA-TION-1	PIC S9(9) Comp	Integer*4	Long
	ALLOCA-TION-2	PIC S9(9) Comp	Integer*4	Long
	ALLOCA-TION-3	PIC S9(9) Comp	Integer*4	Long
	ALLOCA-TION-4	PIC S9(9) Comp	Integer*4	Long
	FSPEC-1	PIC X(44)	Character*44	Stringx=44
	SHADOW-FSPEC-1	PIC X(44)	Character*44	Stringx=44
	FSPEC-2	PIC X(44)	Character*44	Stringx=44
	SHADOW-FSPEC-2	PIC X(44)	Character*44	Stringx=44
	FSPEC-3	PIC X(44)	Character*44	Stringx=44
	SHADOW-FSPEC-3	PIC X(44)	Character*44	Stringx=44
	FSPEC-4	PIC X(44)	Character*44	Stringx=44
UE-INDEX	INDX	PIC X(30)	Character*30	Stringx=30
	STATUS	PIC X(1)	Character*1	Stringx=1

View name	Column name	COBOL data type	FORTTRAN data type	BASIC data type
UE-INDEX-FILE-SPEC	FSPC	PIC X(30)	Character*30	Stringx=30
	CORRUPT-ACTION	PIC X(1)	Character*1	Stringx=1
	NULL-SORTING	PIC X(1)	Character*1	Stringx=1
	READ-VERIFY	PIC X(1)	Character*1	Stringx=1
	FILE-SPEC	PIC X(44)	Character*44	Stringx=44
	SHADOW-FILE-SPEC	PIC X(44)	Character*44	Stringx=44
UE-LOGICAL-DATA-ITEM	LGDI	PIC X(30)	Character*30	Stringx=30
UE-LOGICAL-VIEW	LV	PIC X(30)	Character*30	Stringx=30
	STATUS	PIC X(1)	Character*1	Stringx=1
	BOUND	PIC X(1)	Character*1	Stringx=1
	DATE	PIC X(8)	Character*8	Stringx=8
	LV-BOUND-SIZE	PIC S9(9) Comp	Integer*4	Long
	TIME	PIC X(6)	Character*6	Stringx=6
UE-PROGRAM	PROG	PIC X(30)	Character*30	Stringx=30
	STATUS	PIC X(1)	Character*1	Stringx=1
	DATE-ENROLLED	PIC X(8)	Character*8	Stringx=8
	TIME-ENROLLED	PIC X(6)	Character*6	Stringx=6
	MOD-LEVEL	PIC S9(4) Comp	Integer*2	Word
	LANGUAGE	PIC X(1)	Character*1	Stringx=1

View name	Column name	COBOL data type	FORTRAN data type	BASIC data type
UE-RECORD	RECD	PIC X(30)	Character*30	Stringx=30
	TYPE	PIC S9(4) Comp	Integer*2	Word
	CODE	PIC X(2)	Character*2	Stringx=2
	NO-OF-DATA-ITEMS	PIC S9(4) Comp	Integer*2	Word
	LENGTH	PIC S9(4) Comp	Integer*2	Word
	REDEFINED-DI	PIC X(8)	Character*8	Stringx=8
UE-RMS-DATA-SET	DSET	PIC X(30)	Character*30	Stringx=30
	LENGTH	PIC S9(4) Comp	Integer*2	Word
	TYPE	PIC S9(4) Comp	Integer*2	Word
	STATUS	PIC X(1)	Character*1	Stringx=1
UE-RMS-FILE-SPEC	FSPC	PIC X(30)	Character*30	Stringx=30
	BUCKET-SIZE	PIC S9(4) Comp	Integer*2	Word
	FSPEC-1	PIC X(44)	Character*44	Stringx=44
UE-RMS-KEY	RMSK	PIC X(30)	Character*30	Stringx=30
	NUMBER	PIC S9(9) Comp	Integer*4	Long
	LENGTH	PIC S9(9) Comp	Integer*4	Long
	UNIQUE-KEY	PIC X(1)	Character*1	Stringx=1
	CAN-BE-CHANGED	PIC X(1)	Character*1	Stringx=1

View name	Column name	COBOL data type	FORTTRAN data type	BASIC data type
UE-SECONDARY-KEY	SKEY	PIC X(30)	Character*30	Stringx=30
	UNIQUE-KEY	PIC X(1)	Character*1	Stringx=1
	KEY-DIRECTION	PIC X(1)	Character*1	Stringx=1
	POINTER-ORDERING	PIC X(1)	Character*1	Stringx=1
	POINTER-TYPE	PIC X(1)	Character*1	Stringx=1
	DATA-TYPE-SORT	PIC X(1)	Character*1	Stringx=1
	DUPL-ALLOWED	PIC S9(4) Comp	Integer*2	Word
	STATUS	PIC X(1)	Character*1	Stringx=1
UE-SYSTEM-LOG	FLOG	PIC X(30)	Character*30	Stringx=30
	BLOCK-SIZE	PIC S9(4) Comp	Integer*2	Word
	NO-OF-BLOCKS	PIC S9(9) Comp	Integer*4	Long
	FSPEC-1	PIC X(44)	Character*44	Stringx=44
	FSPEC-2	PIC X(44)	Character*44	Stringx=44
	SHADOW-FSPEC-1	PIC X(44)	Character*44	Stringx=44
	SHADOW-FSPEC-2	PIC X(44)	Character*44	Stringx=44
UE-TASK-LOG	TLOG	PIC X(30)	Character*30	Stringx=30
	BLOCK-SIZE	PIC S9(4) Comp	Integer*2	Word
	NO-OF-BLOCKS	PIC S9(9) Comp	Integer*4	Long
	NO-OF-BUFFERS	PIC S9(4) Comp	Integer*2	Word
	FSPEC-1	PIC X(44)	Character*44	Stringx=44
	SHADOW-FSPEC-1	PIC X(44)	Character*44	Stringx=44

View name	Column name	COBOL data type	FORTRAN data type	BASIC data type
UE-USER	USER	PIC X(30)	Character*30	Stringx=30
	AUTHORITY	PIC S9(4) Comp	Integer*2	Word
	PASSWORD	PIC X(8)	Character*8	Stringx=8
UE-VALIDATION-TABLE	VTBL	PIC X(30)	Character*30	Stringx=30
	STATUS	PIC X(1)	Character*1	Stringx=1
UN-LOGICAL-VIEW	LV	PIC X(30)	Character*30	Stringx=30
	LINE	PIC X(72)	Character*72	Stringx=72
UR-PHYSICAL-LOGICAL-DATA-ITEM	LGDI-NAME	PIC X(30)	Character*30	Stringx=30
	UDD2-NMPM	PIC X(30)	Character*30	Stringx=30
UR-VIEW-TO-VIEW	LV	PIC X(30)	Character*30	Stringx=30
	NAME	PIC X(26)	Character*26	Stringx=26
	SOURCE-NAME	PIC X(26)	Character*26	Stringx=26
	TYPE	PIC S9(4) Comp	Integer*2	Word
	LOGICAL-VIEW-D	PIC X(30)	Character*30	Stringx=30
US-DERIVED-VIEW-CONSTANT	LV	PIC X(30)	Character*30	Stringx=30
	VALUE	PIC X(24)	Character*24	Stringx=24
	NAME	PIC X(26)	Character*26	Stringx=26
	LOGICAL-VIEW	PIC X(30)	Character*30	Stringx=30
UX-BUFFER-OF-DATA-SET	DSET	PIC X(30)	Character*30	Stringx=30
	BUFFER	PIC X(30)	Character*30	Stringx=30
UX-BUFFER-OF-DBDESC	DESC	PIC X(30)	Character*30	Stringx=30
	BUFFER	PIC X(30)	Character*30	Stringx=30
UX-DATA-ITEM-IN-SEC-KEY	SKEY	PIC X(30)	Character*30	Stringx=30
	DATA-ITEM	PIC X(30)	Character*30	Stringx=30

View name	Column name	COBOL data type	FORTTRAN data type	BASIC data type
UX-DATA-ITEM-OF-RECORD	RECD	PIC X(30)	Character*30	Stringx=30
	DATA-ITEM	PIC X(30)	Character*30	Stringx=30
UX-DATA-SET-OF-DBDESC	DESC	PIC X(30)	Character*30	Stringx=30
	DATA-SET	PIC X(30)	Character*30	Stringx=30
UX-DERIVED-VIEW-CONSTANT	LV	PIC X(30)	Character*30	Stringx=30
	VALUE	PIC X(24)	Character*24	Stringx=24
	NAME	PIC X(26)	Character*26	Stringx=26
	LOGICAL-VIEW	PIC X(30)	Character*30	Stringx=30
UX-DOMAIN-OF-DATA-ITEM	DITM	PIC X(30)	Character*30	Stringx=30
	DOMAIN	PIC X(30)	Character*30	Stringx=30
UX-DOMAIN-USES-VALIDATION-TABL	DOMN	PIC X(30)	Character*30	Stringx=30
	VALIDATION-TABLE	PIC X(30)	Character*30	Stringx=30
UX-FILE-SPEC-OF-DATA-SET	DSET	PIC X(30)	Character*30	Stringx=30
	FILE-SPEC	PIC X(30)	Character*30	Stringx=30
UX-FILE-SPEC-OF-DATABASE	DESC	PIC X(30)	Character*30	Stringx=30
	FILE-SPEC	PIC X(30)	Character*30	Stringx=30
UX-FILE-SPEC-OF-INDEX	INDX	PIC X(30)	Character*30	Stringx=30
	FILE-SPEC	PIC X(30)	Character*30	Stringx=30
UX-INDEX-IN-DATA-SET	DSET	PIC X(30)	Character*30	Stringx=30
	INDEX-FILE	PIC X(30)	Character*30	Stringx=30
UX-INDEX-OF-DBDESC	DESC	PIC X(30)	Character*30	Stringx=30
	INDEX-FILE	PIC X(30)	Character*30	Stringx=30
UX-LINKPATH-OF-RECORD	RECD	PIC X(30)	Character*30	Stringx=30
	LINKPATH	PIC X(30)	Character*30	Stringx=30
UX-LOGICAL-DATA-ITEM-OF-DATA-S	DSET	PIC X(30)	Character*30	Stringx=30
	DATA-ITEM	PIC X(30)	Character*30	Stringx=30

View name	Column name	COBOL data type	FORTRAN data type	BASIC data type
UX-LOGICAL-DATA-ITEM-OF-LV	LV	PIC X(30)	Character*30	Stringx=30
	ALIAS	PIC X(30)	Character*30	Stringx=30
	TYPE	PIC S9(4) Comp	Integer*2	Word
	VALUE	PIC X(24)	Character*24	Stringx=24
	DATA-ITEM	PIC X(30)	Character*30	Stringx=30
UX-LOGICAL-VIEW-OF-DBDESC	DESC	PIC X(30)	Character*30	Stringx=30
	LOGICAL-VIEW	PIC X(30)	Character*30	Stringx=30
	DATA-ITEM	PIC X(30)	Character*30	Stringx=30
UX-PHYSICAL-TO-LOGICAL-DATA-IT	DITM	PIC X(30)	Character*30	Stringx=30
	DATA-ITEM	PIC X(30)	Character*30	Stringx=30
UX-PROGRAM-USES-LOGICAL-VIEW	PROG	PIC X(30)	Character*30	Stringx=30
	LOGICAL-VIEW	PIC X(30)	Character*30	Stringx=30
UX-RECORD-OF-DATA-SET	DSET	PIC X(30)	Character*30	Stringx=30
	RECORD	PIC X(30)	Character*30	Stringx=30
UX-RMS-KEY-IN-DATA-ITEM	DITM	PIC X(30)	Character*30	Stringx=30
	RMS-KEY	PIC X(30)	Character*30	Stringx=30
UX-RMS-KEY-IN-DATA-SET	DSET	PIC X(30)	Character*30	Stringx=30
	SECONDARY-KEY	PIC X(30)	Character*30	Stringx=30
UX-SEC-KEY-IN-INDEX	INDX	PIC X(30)	Character*30	Stringx=30
	SECONDARY-KEY	PIC X(30)	Character*30	Stringx=30
UX-SUB-DATA-ITEM	DITM	PIC X(30)	Character*30	Stringx=30
	SUB-DATA-ITEM	PIC X(30)	Character*30	Stringx=30
UX-SYSTEM-LOG-OF-DBDESC	DESC	PIC X(30)	Character*30	Stringx=30
	SYSTEM-LOG	PIC X(30)	Character*30	Stringx=30

View name	Column name	COBOL data type	FORTTRAN data type	BASIC data type
UX-TASK-LOG-OF-DBDESC	DESC	PIC X(30)	Character*30	Stringx=30
	TASK-LOG	PIC X(30)	Character*30	Stringx=30
UX-USER-MAY-RUN-PROGRAM	USER	PIC X(30)	Character*30	Stringx=30
	PROGRAM	PIC X(30)	Character*30	Stringx=30
UX-USER-MAY-USE-LOGICAL-VIEW	USER	PIC X(30)	Character*30	Stringx=30
	LOGICAL-VIEW	PIC X(30)	Character*30	Stringx=30
UX-VIEW-TO-VIEW	LV	PIC X(30)	Character*30	Stringx=30
	NAME	PIC X(26)	Character*26	Stringx=26
	SOURCE-NAME	PIC X(26)	Character*26	Stringx=26
	TYPE	PIC S9(4) Comp	Integer*2	Word
	LOGICAL-VIEW-D	PIC X(30)	Character*30	Stringx=30

Index

A

access definitions 104
accessing views 29, 67
authority codes 65

B

base views 99
BP1 24
buffer details
 accessing
 from UE-BUFFER 30
 in UX-BUFFER-OF-DATA-SET
 68
 in UX-BUFFER-OF-DBDESC
 69

C

cluster support 38
code translation 15
coded portion of a related data
 set 32
coded record
 accessing from UE-RECORD
 54
 identified in UE-DATA-SET 35
CODED-DATA
 column in UX-RECORD-OF-
 DATA-SET 90
 key of UE-DATA-ITEM 32, 83
 key of UE-RECORD 54
 key of UX-DATA-ITEM-OF-
 RECORD 72
comment screen 102
connecting views *See* view
 connecting

constant value
 accessing with two keys 108
 provided in UX-DERIVED-
 VIEW-CONSTANT 74
CSI_NODIRECTORY TRUE 14
CSI_SCHEMA 14

D

data item details
 accessing
 from UE-DATA-ITEM 31
 in UX-DATA-ITEM-IN-SEC-KEY
 70
 in UX-DATA-ITEM-OF-
 RECORD 71
 in UX-DATA-SET-OF-DBDESC
 73
data set details
 accessing
 from UE-DATA-SET 34
 from UE-RMS-DATA-SET 55
data type
 descriptions 109
 format codes 40
database details
 accessing from UE-
 DATABASE-DESCRIPTION
 36
DATABASE-DESCRIPTIONS
 signing on
 to DBAID 25
 to MANTIS with the VIEW
 statement 26
 to SPECTRA 24
 user name to access Directory
 Views 23
DBAID
 accessing Directory Views from
 25
 commands 25
 example command file 14
 example output 15
 signing on 25
 Directory views
 structure of 28

Directory Views

accessing

from DBAID 25

from MANTIS 26

from SPECTRA 24

choosing views to use 27

descriptions of 15

entities 67

format of the descriptions 15

global view file 14

logical names needed 14

using effectively 27

views accessing entity details

29

DIRVIEWS.COM

executing 25

location 14

DIRVIEWS.EXP

BP1 14, 24

importing into SPECTRA 24

location 14

DIRVIEWS.GBL 14

domain details

accessing from UE-DOMAIN 39

in UX-DOMAIN-OF-DATA-ITEM

75

E

entities

accessing 67

defined on the Directory 13

general information about 29

errors at sign on 15

executing

a SPECTRA process 24

DIRVIEWS.COM 25

F

file specification details

accessing

from RMS and PDM data sets

57

from UE-FILE-SPEC 43

from UE-RMS-FILE-SPEC 57

from UX-FILE-SPEC-OF-
DATABASE 78in UX-FILE-SPEC-OF-DATA-
SET 77in UX-FILE-SPEC-OF-INDEX
79**FILE-SPEC**as part of key to UE-FILE-
SPEC 43as part of key to UE-INDEX-
FILE-SPEC 47

format codes 40

function codes 40

FUNCTION-LOGcolumn in UX-SYSTEM-LOG-
OF-DBDESC 95

key of UE-SYSTEM-LOG 63

G

GET command 25

global section type 38

global view file

setting up 14

GO

DBAID command 25

SPECTRA command 24

GVSCHEMA 14

I

importing DIRVIEWS.EXP 24

index details

accessing from UE-INDEX 45

in UX-INDEX-IN-DATA-SET 80

in UX-INDEX-OF-DBDESC 81

indices 45

Indices 79

L

language codes 52
 linkpath
 accessing from UX-LINKPATH-OF-RECORD 82
 alternative 104
 as key to UE-DATA-ITEM 32
 in UX-DATA-ITEM-OF-RECORD 72
 related data set 31
 list processes (LP) 24
 logging on
 possible causes of failure 15
 to DBAID 25
 to SPECTRA 24
 logical data item details 84
 logical data item name
 equivalents 88
 logical data item names
 accessing
 from UE-LOGICAL-DATA-ITEM 48
 logical name definitions
 CSI NODIRECTORY TRUE 14
 CSI_SCHEMA 14
 Directory Views access 14
 GVSCHEMA 14
 setting up 14
 SUPRA_REPORT 14
 LP See list processes

M

MANTIS
 accessing Directory Views from 26
 using the VIEW statement to sign on 26

N

network support 38

P

passing parameters 38
 password
 encrypted output from UE-USER 65
 for DATABASE-DESCRIPTIONS when used from MANTIS 26
 PDM data sets
 accessing
 from UE-RMS-DATA-SET 55
 from UE-RMS-FILE-SPEC 57
 physical file attributes 46
 physical records
 accessing from UE-RECORD 53
 details 53
 PRIMARY-DATA
 column in UX-RECORD-OF-DATA-SET 90
 key of UE-DATA-ITEM 31, 83
 key of UX-DATA-ITEM-OF-RECORD 72
 program details
 accessing
 from UE-PROGRAM 51
 from UX-USER-MAY-RUN-PROGRAM 97

R

record details
 accessing from UX-RECORD-OF-DATA-SET 90
 related data set
 linkpath 31
 Related data set
 coded portion 32
 RELATED-DATA
 as column in UX-RECORD-OF-DATA-SET 90
 as part of key to UE-DATA-ITEM 31, 83
 as part of key to UE-RECORD 54
 as part of key to UX-DATA-ITEM-OF-RECORD 72

relationships between entities 13
reports 24
RMS data sets
 accessing
 from UE-RMS-DATA-SET 55
 from UE-RMS-FILE-SPEC 57
RMS key details
 accessing
 from UE-RMS-KEY 59
 from UX-RMS-KEY-IN-DATA-
 ITEM 91
 from UX-RMS-KEY-IN-DATA-
 SET 92
RMS-DATA 72, 90
RMS-KEY
 column in UX-RMS-KEY-IN-
 DATA-ITEM 91
 column in UX-RMS-KEY-IN-
 DATA-SET 92

S

secondary key details
 accessing
 from UE-SECONDARY-KEY
 61
 from UX-SEC-KEY-IN-INDEX
 93
SHADOW-OPTION codes 37
SHOW command 24
sign codes 41
signing on
 possible causes of failure 15
 to DBAID 25
 to SPECTRA 24
single-task database 37
special views 15
SPECTRA
 accessing Directory Views from
 24
 executing a process 24

status codes
 in UE-DATABASE-
 DESCRIPTION 37
 in UE-DATA-SET 35
 in UE-DOMAIN 42
 in UE-LOGICAL-VIEW 50
 in UE-PROGRAM 51
 in UE-RMS-DATA-SET 56
sub-data-item details 94
subdefined data items 94
SUPRA directory
 structure of 27
SUPRA Directory View
 relationships 27
SUPRA_REPORT 14
SUPRAD 14
system log details
 accessing
 from UE-SYSTEM-LOG 63
 from UX-SYSTEM-LOG-OF-
 DBDESC 95
 from UX-TASK-LOG-OF-
 DBDESC 96
SYSTEM-LOG
 column in UX-SYSTEM-LOG-
 OF-DBDESC 95
 description in UE-SYSTEM-
 LOG 63

T

task log details
 accessing from UE-TASK-LOG
 64
TASK-LOG
 column in UX-TASK-LOG-OF-
 DBDESC 96
 description in UE-TASK-LOG
 64
translating codes 15
type codes
 in UE-BUFFER 30
 in UE-DATA-ITEM 32
 in UE-RECORD 54
 in UE-RMS-DATA-SET 55
 in UR-VIEW-TO-VIEW 107
 in UX-LOGICAL-DATA-ITEM-
 OF-LV 86

U

- UC-BUFFER
 - data type description 109
 - general information about 16
- UC-DATABASE-DESCRIPTION
 - comment view 102
 - data type description 109
 - general information about 16
- UC-DATA-ITEM
 - data type description 109
 - general information about 16
- UC-DATA-SET
 - data type description 109
 - general information about 16
- UC-DOMAIN
 - data type description 109
 - general information about 16
- UC-FILE-SPEC
 - data type description 109
 - general information about 16
- UC-INDEX
 - data type description 109
 - general information about 16
- UC-INDEX-FILE-SPEC
 - data type description 109
 - general information about 16
- UC-LOGICAL-DATA-ITEM
 - data type description 110
 - general information about 16
- UC-LOGICAL-VIEW
 - data type description 110
 - general information about 16
- UC-PROGRAM
 - data type description 110
 - general information about 16
- UC-RMS-DATA-SET
 - data type description 110
 - general information about 16
- UC-RMS-FILE-SPEC
 - data type description 110
 - general information about 16
- UC-RMS-KEY
 - data type description 110
 - general information about 17
- UC-SECONDARY-KEY
 - data type description 110
 - general information about 17
- UC-SYSTEM-LOG
 - data type description 110
 - general information about 17
- UC-TASK-LOG
 - data type description 110
 - general information about 17
- UC-USER
 - data type description 110
 - general information about 17
- UC-VALIDATION-TABLE
 - data type description 110
 - general information about 17
- UE-BUFFER
 - accessing
 - buffer details 30
 - from UX-BUFFER-OF-DATA-SET 68
 - data type description 110
 - general information about 17
 - type codes 30
- UE-DATABASE-DESCRIPTION
 - accessing database details 36
 - data type description 112
 - general information about 17
 - status codes 37
- UE-DATA-ITEM
 - accessing
 - data item details 31
 - from UX-DATA-ITEM-IN-SEC-KEY 70
 - from UX-LINKPATH-OF-RECORD 82
 - data type description 111
 - general information about 17
 - type codes 32
 - use codes 32

UE-DATA-SET

- accessing
 - data set details 34
 - from UX-DATA-SET-OF-DBDESC 73
- data type description 111
- general information about 17
- status codes 35

UE-DOMAIN

- accessing
 - domain details 39
 - from UX-DOMAIN-OF-DATA-ITEM 75
- data type description 113
- format codes 40
- function codes 40
- general information about 17
- status codes 42

UE-FILE-SPEC

- accessing
 - file specification details 43
 - from UX-FILE-SPEC-OF-DATABASE 78
 - from UX-FILE-SPEC-OF-DATA-SET 77
- data type description 114
- description 17

UE-INDEX

- accessing
 - from UX-FILE-SPEC-OF-INDEX 79
 - from UX-INDEX-IN-DATA-SET 80
 - from UX-INDEX-OF-DBDESC 81
 - index details 45
- data type description 114
- general information about 17

UE-INDEX-FILE-SPEC

- accessing physical file attribute details 46
- data type description 115
- general information about 18

UE-LOGICAL-DATA-ITEM

- accessing
 - from UX-LOGICAL-DATA-ITEM-OF-DATA-S 84
 - from UX-LOGICAL-DATA-ITEM-OF-LV 85
 - from UX-PHYSICAL-TO-LOGICAL-DATA-IT 88
 - logical data item names 48
- data type description 115
- general information about 18

UE-LOGICAL-VIEW

- accessing
 - from UX-LOGICAL-VIEW-OF-DBDESC 87
 - from UX-PROGRAM-USES-LOGICAL-VIEW 89
 - view details 49
- data type description 115
- general information about 18
- status codes 50

UE-PROGRAM

- accessing
 - from UX-USER-MAY-RUN-PROGRAM 97
 - program details 51
- data type description 115
- general information about 18
- language codes 52
- status codes 51

UE-RECORD

- accessing
 - from UX-RECORD-OF-DATA-SET 90
 - physical record details 53
- data type description 116
- general information about 18
- type codes 54

UE-RMS-DATA-SET

- accessing data set details 55
- data type description 116
- general information about 18
- status codes 56
- type codes 55

- UE-RMS-FILE-SPEC
 - accessing file specification details 57
 - data type description 116
 - general information about 18
- UE-RMS-KEY
 - accessing
 - from UX-RMS-KEY-IN-DATA-ITEM 91
 - RMS key details 59
 - data type description 116
 - general information about 18
- UE-SECONDARY-KEY
 - accessing
 - from UX-SECONDARY-KEY 93
 - secondary key details 61
 - data type description 117
 - general information about 18
- UE-SYSTEM-LOG
 - accessing system log details 63
 - data type description 117
 - general information about 19
- UE-TASK-LOG
 - accessing
 - from UX-TASK-LOG-OF-DBDESC 96
 - task log details 64
 - data type description 117
 - general information about 19
- UE-USER
 - accessing user details 65
 - data type description 118
 - general information about 19
- UE-VALIDATION-TABLE
 - accessing
 - from UX-DOMAIN-USES-VALIDATION-TABL 76
 - validation table details 66
 - data type description 118
 - general information about 19
- unit codes 40
- UN-LOGICAL-VIEW
 - access definitions 104
 - data type description 118
 - general information about 19
- UR-PHYSICAL-LOGICAL-DATA-ITEM
 - data type description 118
 - general information about 19
 - showing connections 105
- UR-VIEW-TO-VIEW
 - accessing where-used details 106
 - data type description 118
 - general information about 19
 - type codes 107
- US-DERIVED-VIEW-CONSTANT
 - data type description 118
 - general information about 19
- use codes 32
- user database not in directory
 - error message 15
- user details
 - accessing
 - from UE-USER 65
 - accessing from UX-USER-MAY-USE-LOGICAL-VIEW 98
 - user name
 - signing on to SPECTRA 23
- UX-BUFFER-OF-DATA-SET
 - data type description 118
 - general information about 19
 - providing buffer details 68
- UX-BUFFER-OF-DBDESC
 - accessing from UX-BUFFER-OF-DBDESC 69
 - data type description 118
 - general information about 19
 - providing buffer details 69
- UX-DATA-ITEM-IN-SEC-KEY
 - data type description 118
 - general information about 20
 - providing data item details in secondary key 70

- UX-DATA-ITEM-OF-RECORD
 - data type description 119
 - general information about 20
 - providing data item details 71
- UX-DATA-SET-OF-DBDESC
 - data type description 119
 - general information about 20
 - providing data item details 73
- UX-DERIVED-VIEW-CONSTANT
 - data type description 119
 - general information about 20
 - providing constant value 74
- UX-DOMAIN-OF-DATA-ITEM
 - data type description 119
 - general information about 20
 - providing domain details 75
- UX-DOMAIN-USES-VALIDATION-TABL
 - data type description 119
 - general information about 20
 - providing validation table details 76
- UX-FILE-SPEC-OF-DATABASE
 - accessing file specification details 78
 - data type description 119
 - general information about 20
- UX-FILE-SPEC-OF-DATA-SET
 - accessing file specification details 77
 - data type description 119
 - general information about 20
- UX-FILE-SPEC-OF-INDEX
 - accessing file specification details 79
 - data type description 119
 - general information about 20
- UX-INDEX-IN-DATA-SET
 - accessing index details 80
 - data type description 119
 - general information about 20
- UX-INDEX-OF-DBDESC
 - accessing index details 81
 - data type description 119
 - general information about 20
- UX-LINKPATH-OF-RECORD
 - accessing linkpath 82
 - data type description 119
 - general information about 21
- UX-LOGICAL-DATA-ITEM-OF-DATA-S
 - accessing logical data item details 84
 - data type description 119
 - general information about 21
- UX-LOGICAL-DATA-ITEM-OF-LV
 - accessing logical data item details 85
 - data type description 120
 - general information about 21
 - type codes 86
- UX-LOGICAL-VIEW-OF-DBDESC
 - accessing views 87
 - data type description 120
 - general information about 21
- UX-PHYSICAL-TO-LOGICAL-DATA-IT
 - accessing logical data item names equivalents 88
 - data type description 120
 - general information about 21
- UX-PROGRAM-USES-LOGICAL-VIEW
 - accessing views used by a program 89
 - data type description 120
 - general information about 21
- UX-RECORD-OF-DATA-SET
 - accessing record details 90
 - data type description 120
 - general information about 21
- UX-RMS-KEY-IN-DATA-ITEM
 - accessing RMS key details 91
 - data type description 120
 - general information about 21

UX-RMS-KEY-IN-DATA-SET
 accessing RMS key details 92
 data type description 120
 general information about 21
UX-SEC-KEY-IN-INDEX
 accessing secondary key
 details 93
 data type description 120
 general information about 22
UX-SUB-DATA-ITEM
 data type description 120
 displaying sub-data-item details
 94
 general information about 22
UX-SYSTEM-LOG-OF-DBDESC
 accessing system log details 95
 data type description 120
 general information about 22
UX-TASK-LOG-OF-DBDESC
 accessing system log details 96
 data type description 121
 general information about 22
UX-USER-MAY-RUN-PROGRAM
 accessing
 program details 97
 data type description 121
 general information about 22
**UX-USER-MAY-USE-LOGICAL-
VIEW**
 data type description 121
 displaying views available to a
 specified user 98
 general information about 22
UX-VIEW-TO-VIEW
 accessing base views 99
 data type description 121
 general information about 22

V

validation option codes 41
 validation table details
 accessing from UE-
 VALIDATION-TABLE 66
 in UX-DOMAIN-USES-
 VALIDATION-TABL 76
VAX cluster support 38
 view connecting
 UE-DATABASE and UE-DATA-
 SET 73
 UE-DATABASE and UE-INDEX
 81
 UE-DATABASE-DESCRIPTION
 and UE-BUFFER 69
 UE-DATABASE-DESCRIPTION
 and UE-FILE-SPEC 78
 UE-DATABASE-DESCRIPTION
 and UE-LOGICAL-VIEW 87
 UE-DATABASE-DESCRIPTION
 and UE-SYSTEM-LOG 95
 UE-DATABASE-DESCRIPTION
 and UE-TASK-LOG 96
 UE-DATA-ITEM and UE-
 DOMAIN 75
 UE-DATA-ITEM and UE-
 LOGICAL-DATA ITEM 88
 UE-DATA-ITEM and UE-RMS-
 KEY 91
 UE-DATA-SET and UE-
 BUFFER 68
 UE-DATA-SET and UE-FILE-
 SPEC 77
 UE-DATA-SET and UE-INDEX
 80
 UE-DATA-SET and UE-
 LOGICAL-DATA-ITEM 84
 UE-DATA-SET and UE-
 RECORD 90
 UE-DATA-SET and UE-RMS-
 KEY 92
 UE-DOMAIN and UE-
 VALIDATION-TABLE 76

view connecting (*cont.*)
 UE-INDEX and UE-FILE-SPEC
 79
 UE-INDEX and UE-
 SECONDARY-KEY 93
 UE-LOGICAL-VIEW and UE-
 LOGICAL-DATA-ITEM 85
 UE-PROGRAM and UE-
 LOGICAL-VIEW 89
 UE-RECORD and UE-DATA-
 ITEM 71
 UE-RECORD and UE-DATA-
 ITEM' 82
 UE-SECONDARY-KEY and
 UE-DATA-ITEM 70
 UE-USER and UE-LOGICAL-
 VIEW 98
 UE-USER and UE-PROGRAM
 97
view details
 accessing
 from UE-LOGICAL-VIEW 49

VIEW statement 26
views
 accessing
 comments 102
 definitions 104
 entity details 29
 from UX-LOGICAL-VIEW-OF-
 DBDESC 87
 from UX-PROGRAM-USES-
 LOGICAL-VIEW 89
 relationships between entities
 67
 for special purposes 101
 to show alternative relationship
 entities 104
 using two keys to access
 constant values 108

W

where-used details 106