IBM

DataRefresher

Version 1

MVS and VM User’s Guide

Part of the Information Warehouse family
First Edition (October 1994)

This edition applies to Version 1 of DataRefresher, Program Number 5696-703, and to all releases until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

A form for readers' comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Software Solutions,
Information Development (IISL),
2 Burlington Road,
Dublin 4,
Ireland.
Fax: (Ireland) +353 - 1 -6614246
IBMMail: IEIBM3FL at IBMMail
INTERNET: IEIBM3FL@IBMMail.COM

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1986, 1994. All rights reserved.
Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.
## Contents

Notices .................................................. vii  
Authorized use of IBM online books .................. vii  
Trademarks and Service Marks ....................... viii  

About this book ........................................ ix  
What you should know already ....................... ix  
DataRefresher library overview ..................... ix  
Other information ..................................... x  

### Part 1. Introducing DataRefresher

Chapter 1. DataRefresher overview ........................ 3  
DataRefresher and MVS .................................... 4  
DataRefresher and VM .................................... 4  
Valid DataRefresher data sources and target systems .... 4  
  Data sources ........................................... 6  
  Extending the range of data sources ................ 6  
  Target systems ........................................ 6  
  Extending the range of targets ..................... 7  
  Manipulating the extracted data .................... 7  

Chapter 2. How DataRefresher works ..................... 9  
Step 1. Describing the source data ................... 10  
Step 2. Creating an extract request ................. 12  
Step 3. Running the extract manager .................. 13  

Chapter 3. How to create data descriptions .......... 15  
Overview of tasks ....................................... 15  
CREATE DATATYPE ...................................... 16  
  Specifying source and target data ................ 16  
  Examples of the CREATE DATATYPE command ........ 17  
CREATE DXFILE ......................................... 18  
  Assigning names in a DXFILE description .......... 19  
  Describing the organization of the file .......... 19  
  Specifying user-written exit routines .............. 20  
  Examples of the CREATE DXFILE command .......... 20  
CREATE DXPSB ........................................... 24  
  Assigning names in your DXPSB description ....... 25  
  Describing the organization of the DXPSB and DXPCB . 25  
  Examples of the CREATE DXPSB command ............ 31  
CREATE DXVIEW .......................................... 39  
  Assigning names in DXVIEW descriptions .......... 40  
  Assigning types of DXVIEW descriptions .......... 40  
  Examples of CREATE DXVIEW ......................... 41  

Chapter 4. Creating data descriptions and extract requests .... 49  
Structures Access Program ............................. 49  
Dictionary Access Program ............................ 49  
DataRefresher dialogs ................................. 49  

© Copyright IBM Corp. 1986, 1994
| Administrative Dialogs                                | 50 |
| End User Dialogs                                      | 50 |
| Online DataRefresh commands                           | 51 |
| System editor                                         | 52 |
| Extracting data from an IMS database                  | 52 |
| Extracting data from a VSAM data set                  | 53 |
| Extracting data from a DB2 database                   | 53 |
| Extracting data from an SQL/DS database               | 54 |
| Extracting data from an IMS database and a DB2 database| 54 |
| Extracting data from a DB2 database and a VSAM data set| 55 |
| Extracting data from an IMS database and a VSAM data set| 56 |

### Part 2. Using the Administrative Dialogs

| Chapter 5. What are Administrative Dialogs?                | 59 |
| Working with Administrative Dialogs                       | 59 |
| How you can use the Administrative Dialogs                | 60 |
| Navigating through Administrative Dialogs                  | 61 |

| Chapter 6. Starting the Administrative Dialogs             | 63 |
| Starting an MVS dialog session                             | 63 |
| Starting a VM dialog session                               | 63 |

| Chapter 7. Maintaining your JCL and JCS files              | 65 |
| Creating and editing JCL/JCS models                        | 66 |
| JCL and JCS models supplied with DataRefresh              | 67 |
| Directing output in MVS                                   | 69 |
| Directing output in VM                                    | 70 |

| Chapter 8. Extracting data from a non-relational source    | 73 |
| Step 1. Creating JCL using a model                         | 73 |
| Step 2. Creating a DXTPSB or DXTFILE description using a model | 75 |
| Step 3. Creating a DXTVIEW using a model                   | 77 |
| Step 4. Sending data descriptions to the FDTLIB            | 77 |
| Step 5. Creating JCS using a model                         | 78 |
| Step 6. Creating an extract request using a model          | 79 |
| Step 7. Sending an extract request and JCS to the EXTLIB   | 80 |
| Step 8. Creating a job to run the DEM                      | 81 |
| Step 9. Running the DEM to process the extract             | 83 |

| Chapter 9. Extracting data from a relational source        | 85 |
| Step 1. Creating JCS using a model                         | 85 |
| Step 2. Creating an extract request using a model          | 86 |
| Step 3. Creating a job to run the REM                       | 87 |
| Step 4. Running the REM to process the extract             | 88 |

<p>| Chapter 10. Extracting data using the SAP                  | 89 |
| Conventions and restrictions                              | 89 |
| Conventions for SAP data structure conversion             | 89 |
| Restrictions to SAP processing                            | 90 |
| Specifying SAP execution information                      | 92 |
| Starting the SAP                                          | 92 |
| Using the SAP Dialog panels                               | 92 |</p>
<table>
<thead>
<tr>
<th>Content</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Editing the skeleton JCL</td>
<td>97</td>
</tr>
<tr>
<td>Submitting the JCL for batch processing</td>
<td>102</td>
</tr>
<tr>
<td>Using SAP output</td>
<td>102</td>
</tr>
<tr>
<td>Editing SAP output</td>
<td>103</td>
</tr>
<tr>
<td>Invoking UIM to store SAP output</td>
<td>103</td>
</tr>
<tr>
<td>Importing SAP output into the DataRefresher dialog library</td>
<td>103</td>
</tr>
<tr>
<td>Error handling</td>
<td>104</td>
</tr>
<tr>
<td>Chapter 11. Creating data descriptions using the DAP</td>
<td>107</td>
</tr>
<tr>
<td>Step 1. Creating the input for the DAP</td>
<td>107</td>
</tr>
<tr>
<td>Using the EXECUTE command</td>
<td>107</td>
</tr>
<tr>
<td>Using the DAP in batch</td>
<td>107</td>
</tr>
<tr>
<td>Using the DAP from the Administrative Dialogs</td>
<td>110</td>
</tr>
<tr>
<td>Step 2. Editing the DAP output</td>
<td>112</td>
</tr>
<tr>
<td>DDPUNCH output</td>
<td>112</td>
</tr>
<tr>
<td>DDLIST output</td>
<td>113</td>
</tr>
<tr>
<td>Errors not detected by the DAP</td>
<td>113</td>
</tr>
<tr>
<td>Chapter 12. Maintaining data descriptions</td>
<td>115</td>
</tr>
<tr>
<td>Deleting a data description</td>
<td>115</td>
</tr>
<tr>
<td>Printing a data description</td>
<td>117</td>
</tr>
<tr>
<td>Chapter 13. Maintaining extract requests</td>
<td>121</td>
</tr>
<tr>
<td>Checking the status of an extract request</td>
<td>121</td>
</tr>
<tr>
<td>Deleting an extract request from an EXTLIB</td>
<td>123</td>
</tr>
<tr>
<td>Listing extract requests in an EXTLIB</td>
<td>124</td>
</tr>
<tr>
<td>Chapter 14. Changing the default profile</td>
<td>127</td>
</tr>
<tr>
<td>Dialogs Profile Options panel 1</td>
<td>127</td>
</tr>
<tr>
<td>Dialogs Profile Options panel 2</td>
<td>128</td>
</tr>
<tr>
<td>Dialogs Profile Options panel 3</td>
<td>129</td>
</tr>
<tr>
<td>Dialogs Profile Options panel 4</td>
<td>131</td>
</tr>
<tr>
<td>Dialogs Profile Options panel 5</td>
<td>132</td>
</tr>
<tr>
<td>Dialogs Profile Options panel 6</td>
<td>133</td>
</tr>
<tr>
<td>Dialogs Profile Options panel 7</td>
<td>134</td>
</tr>
<tr>
<td>Chapter 15. Providing access to dialog objects</td>
<td>135</td>
</tr>
<tr>
<td>Providing edited JCL files</td>
<td>135</td>
</tr>
<tr>
<td>Using object sharing</td>
<td>137</td>
</tr>
<tr>
<td>Using object sharing and maintaining a personal DataRefresher library</td>
<td>138</td>
</tr>
<tr>
<td>Chapter 16. Setting up End User Dialogs</td>
<td>139</td>
</tr>
<tr>
<td>Building or updating the nickname table</td>
<td>140</td>
</tr>
<tr>
<td>Updating the end user table</td>
<td>141</td>
</tr>
<tr>
<td>Requesting source table description data</td>
<td>142</td>
</tr>
<tr>
<td>Editing a UIM job to build or update the master index table (MIT)</td>
<td>145</td>
</tr>
<tr>
<td>Editing a REM job to build or update the master index table (MIT)</td>
<td>146</td>
</tr>
<tr>
<td>Maintaining the master index table (MIT)</td>
<td>148</td>
</tr>
<tr>
<td>Providing access to the master index table (MIT)</td>
<td>148</td>
</tr>
<tr>
<td>Enrolling users</td>
<td>149</td>
</tr>
<tr>
<td>Enrolling MVS DataRefresher dialogs users</td>
<td>149</td>
</tr>
<tr>
<td>Enrolling VM DataRefresher dialogs users</td>
<td>151</td>
</tr>
</tbody>
</table>
Part 3. Using the End User Dialogs ................................. 155

Chapter 17. What are End User Dialogs? .......................... 157
Working with End User Dialogs .................................. 157
Navigating through the End User Dialogs ...................... 157
End User Dialogs function keys ................................ 159
Getting help ...................................................... 161
Using nicknames .................................................. 161
Setting up End User Dialogs ..................................... 162

Chapter 18. Starting the End User Dialogs ...................... 163

Chapter 19. Extracting data from a non-relational source .... 165
Task 1. Building the extract request .............................. 165
  Step 1. Starting an End User Dialogs session ................. 165
  Step 2. Selecting a DXTVIEW ................................ 166
  Step 3. Selecting columns from the DXTVIEW ............... 168
  Step 4. Specifying the source data conditions ............... 168
  Step 5. Identifying the target table .......................... 170
Task 2. Providing database access information .................. 172
Task 3. Saving the extract request .............................. 173
Task 4. Sending the extract request for processing ............ 174

Chapter 20. Extracting from DB2 to SQL/DS .................... 177
Task 1. Building the extract request .............................. 177
  Step 1. Starting the End User Dialogs session ................. 177
  Step 2. Choosing the source tables ............................ 178
  Step 3. Selecting the columns ................................ 180
  Step 4. Specifying conditions ................................ 184
  Step 5. Joining two source columns ............................ 185
  Step 6. Identifying the target table ............................ 187
Task 2. Providing database access information .................. 189
Task 3. Saving the extract request .............................. 190
Task 4. Sending the extract request .............................. 191

Chapter 21. JCL/JCS files and End User Dialogs requests .... 193

Part 4. Appendixes, terms and abbreviations .................... 195

Appendix A. Data set and file definitions ....................... 197
  MVS common data set definitions ............................. 197
  MVS user-specific data set definitions ....................... 197
  VM common file definitions .................................. 198
  VM user-specific file definitions ............................ 198

Terms and abbreviations ......................................... 199

Index .................................................................. 209
Notices

References in this publication to IBM® products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Corporation, IBM Director of Licensing, 208 Harbor Drive, Stamford, Connecticut, United States, 06904.

Authorized use of IBM online books

For online versions of this book, we authorize you to:

Copy, modify, and print the documentation contained on the media, for use within your enterprise, provided you reproduce the copyright notice, all warning statements, and other required statements on each copy or partial copy.

Transfer the original unaltered copy of the documentation when you transfer the related IBM product (which may be either machines you own, or programs, if the program's license terms permit a transfer). You must, at the same time, destroy all other copies of the documentation.

You are responsible for payment of any taxes, including personal property taxes, resulting from this authorization.

THERE ARE NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

Your failure to comply with the terms above terminates this authorization. Upon termination, you must destroy your machine readable documentation.
## Trademarks and Service Marks

The following terms, denoted by an asterisk (*), used in this publication, are trademarks of the IBM Corporation in the United States and/or other countries:

<table>
<thead>
<tr>
<th>AIX</th>
<th>OS/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS/400</td>
<td>RACF</td>
</tr>
<tr>
<td>DATABASE 2</td>
<td>SQL/DS</td>
</tr>
<tr>
<td>DB2</td>
<td>VM/XA</td>
</tr>
<tr>
<td>DB2/6000</td>
<td>Systems Application Architecture</td>
</tr>
<tr>
<td>DXT</td>
<td></td>
</tr>
<tr>
<td>IBM</td>
<td></td>
</tr>
<tr>
<td>MVS/ESA</td>
<td></td>
</tr>
</tbody>
</table>

The following term, denoted by two asterisks (**), in this publication, are trademarks of other companies:

- Bridge/Fastload (Bridge Technology Inc)
About this book

This book contains information about using DataRefresher to build extract requests in an MVS* or VM* environment. It describes how to use DataRefresher to extract data from DataRefresher data sources and move the extracted data into relational tables, physical sequential data sets, or CMS files. In particular, the book describes how to use the DataRefresher:

- Administrative Dialogs
- End User Dialogs

If you want to use the OS/2 interface provided with DataRefresher, refer to the DataRefresher OS/2 User's Guide.

What you should know already

This book assumes that you understand the structure and function of DataRefresher. It also assumes that you are familiar with the following:

- Conversational Monitor System (CMS)
- IBM DATABASE 2* (DB2*)
- IBM Virtual Machine (VM)
- Information Management System (IMS)
- Interactive System Productivity Facility (ISPF)
- Multiple Virtual Storage (MVS)
- SQL/Data System (SQL/DS™)

DataRefresher library overview

The following describes the contents and organization of information in the DataRefresher Version 1 library.

<table>
<thead>
<tr>
<th>DataRefresher Version 1 (5696-703)</th>
<th>Order number</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>An Introduction</strong></td>
<td>GH19-6993-00</td>
</tr>
<tr>
<td>This book provides an overview of DataRefresher. It describes the uses, benefits, and requirements of DataRefresher to help you evaluate the product.</td>
<td></td>
</tr>
<tr>
<td><strong>Licensed Program Specifications</strong></td>
<td>GH19-9994-00</td>
</tr>
<tr>
<td>This document briefly describes the technical information for DataRefresher and is the warranty for the product.</td>
<td></td>
</tr>
<tr>
<td><strong>Administration Guide</strong></td>
<td>SH19-6995-00</td>
</tr>
<tr>
<td>This book describes how to plan for the installation and the use of DataRefresher in your organization. It describes how to set up and administer DataRefresher.</td>
<td></td>
</tr>
<tr>
<td><strong>MVS and VM User's Guide</strong></td>
<td>SH19-6996-00</td>
</tr>
<tr>
<td>This book describes how to use DataRefresher in an MVS or VM environment. In particular, it describes how to use the DataRefresher Administrative Dialogs and End User Dialogs to create and run extract requests.</td>
<td></td>
</tr>
</tbody>
</table>
DataRefresher Version 1 (5696-703)  

OS/2 User's Guide  
This book describes how to use DataRefresher on a workstation. It describes how to register your host data sources and create an extract which can be run on the host MVS system.  
SH19-6997-00

Exit Routines  
This book describes how to write user exit routines to be used by DataRefresher when an extract is processed.  
SH19-6998-00

Command Reference  
This book provides detailed reference information for all of the DataRefresher commands and procedures.  
SH19-6999-00

Messages and Codes  
This book lists the DataRefresher messages with explanations and suggested responses.  
SC19-5000-00

Diagnosis Guide  
This book contains the information required to diagnose problems with DataRefresher. It also contains information that can help you communicate with the IBM Support Center to isolate and solve problems with DataRefresher.  
LY19-6386-00

Other information

The following books are referenced in this book:

DATABASE 2 publications

IMS publications

ISPF publications
ISPF Dialog Management Guide and Reference (SC34-4266).  
ISPF Dialog Management Services (SC34-2088).

SQL/DS publications
SQL/DS Database Services Utility for IBM VM Systems (SH09-8088).
Part 1. Introducing DataRefresher
Chapter 1. DataRefresher overview

Within modern organizations, database systems can be grouped based on their use into the following:

- **Operational Systems**
  
  An organization's operational systems support the organization's business activities, for example, payroll and accounting systems. Operational systems are developed to perform a fixed set of transactions, and may have their data stored in IMS databases, VSAM data sets, or DB2 databases.

- **Informational (Decision Support) Systems**
  
  These systems are used by an organization to store the information used to generate reports or perform market analysis. Informational systems are less fixed; they provide data required by an organization for monitoring its business performance. This data can change as the requirements, or emphasis, of an organization changes.

Both database systems have different sets of requirements, and so different characteristics and formats.

DataRefresher provides you with capabilities for copying, refining and manipulating data from a source database or file on one system and formatting it for a target database or file on another system. You can use DataRefresher to copy data between:

- Operational systems
- Informational systems
- Informational systems and operational systems

Data can be extracted data from several sources, combined, and the resulting output sent to another system.

If you use a variety of systems and databases, you can use DataRefresher to extract data instead of writing individual application programs. Once an extract request is created, you can submit it, or use it as a model for creating similar extracts.

DataRefresher provides an object-oriented user interface operating in the OS/2 environment that enables you to define several DataRefresher's sources and targets, and to create extracts. It also provides a variety of methods for defining data sources and creating extract requests in an MVS and a VM environment.
**DataRefresher and MVS**

DataRefresher operating under MVS provides you with the following for generating and processing extracts on the full range of MVS DataRefresher data sources:

- Structures Access Program (SAP)
- Dictionary Access Program (DAP)
- Online DataRefresher commands
- DataRefresher dialogs

You can also use your system editor to create your data descriptions and extract requests based on the models supplied with DataRefresher.

The OS/2 GUI is the recommended way of creating non-relational data extracts. The MVS facilities should be used when creating a relational, DB/2, data extract as this type of extract is not supported by the GUI.

To create an extract for a relational data source using DataRefresher you need to:

1. Build an extract request.
2. Send the extract to the Relational Extract Manager (REM) to be run.

As the source data is stored in a relational database (a DB2 database), DataRefresher uses the data descriptions that are stored in the database catalog as a description of the data.

---

**DataRefresher and VM**

The DataRefresher VM feature enables you to use DataRefresher in a VM environment. This feature provides you with the DataRefresher dialogs for creating extracts on Structured Query Language/Data System (SQL/DS*) databases.

As with DataRefresher in an MVS environment, you can use your system editor to create your data descriptions and extract requests based on the models supplied with DataRefresher.

To create an extract for a relational data source using DataRefresher you need to:

1. Build an extract request.
2. Send the extract to the Relational Extract Manager (REM) to be run.

As the source data is stored in a relational database (an SQL/DS database), DataRefresher uses the data descriptions that are stored in the database catalog as a description of the data.

---

**Valid DataRefresher data sources and target systems**

Figure 1 on page 5 provides an overview of all the different types of data source that can be used with DataRefresher. It also shows all of the different types of target systems to which the extracted data can be sent.
Figure 1. Valid DataRefresher Sources and Targets
Data sources

DataRefresher provides you with facilities for extracting data from any of the data sources shown in Figure 1 on page 5:

- Information Management System (IMS) databases
- Virtual Storage Access Method (VSAM) data sets
- Physical sequential data sets (flat files)
- Database 2 (DB2) databases (MVS only)
- Structured Query Language/Data System (SQL/DS) databases (VM only)

Extending the range of data sources

DataRefresher provides you with facilities for registering exit routines which can be used by an extract to extend the range of data sources, or to format the extracted data for a target system. The following types of exit routine can be used to extend the range of data sources:

Generic data interface (GDI) exit routines

GDI exit routines can be developed to access data that is stored in a data source which is not directly supported by DataRefresher. A GDI exit makes it possible for you to extract data from Integration Exchange Format (IXF) files, and from other IBM and non-IBM sources which are not directly supported by DataRefresher.

Map capture exit routines

Map capture exit routines can be developed so that the DataRefresher definition and extract information for a specific extract can be saved and used by DataPropagator NonRelational for data propagation.

DataRefresher provides sample exit routines and interface control blocks for each exit routine.

Target systems

DataRefresher can format extracted data for the following types of targets:

MVS systems
- DB2 tables
- Physical sequential data sets (flat files)
- Physical sequential data sets (in IXF format)

VM systems
- SQL/DS tables
- CMS files
- CMS files (in IXF format)

VSE systems
- SQL/DS tables

AS/400* systems
- Physical data files that are defined when the file arrives

DataRefresher can also be used with other IBM products or vendor supplied products, such as Bridge/Fastload** product, to support the following targets:

OS/2 systems
- DB2/2* tables

AIX* systems
- DB2/6000* tables
Extending the range of targets

Support is provided for other IBM and non-IBM targets by the **Generic Output Interface** (GOI) exit routine. GOI exit routines provide you with the flexibility to make changes to the extracted data, ensuring that the correct data is received by the target system. You can develop a GOI exit to:

- Format the data for a target system that is not directly supported by DataRefresher
- Summarize the extracted data
- Provide totals for the extracted data
- Make changes to the extracted data; for example, add or remove data
- Perform complex data enhancement

Manipulating the extracted data

The following exit routine interfaces are also supplied with DataRefresher. These exit routines can be used to manipulate the extracted data before it is written to a target system:

**Data exit routines**

Data exit routines can be developed to make selected changes to the format of any extracted data before DataRefresher loads it to a target.

**Date/time conversion exit routines**

Date/Time conversion exits can be developed to change data and/or time fields to the International Standards Organization (ISO) format used by DataRefresher.

**User data type exit routines**

User data type exits can be developed to convert data types to a format that is supported by DataRefresher.

**Accounting exit routines**

Accounting exit routines can be developed to keep track of the resources used by the Data Extract Manager (DEM).

DataRefresher provides sample exit routines and interface control blocks for all of the exit routine interfaces supplied with DataRefresher.
Chapter 2. How DataRefresher works

There are three main steps involved in creating and processing an extract from one of the MVS or VM sources. To copy and filter data from one of the sources shown in Figure 1 on page 5, you have to:

1. Describe the data to be extracted, and store the description in the File Description Table Library (FDTLIB).

   When copying data using DataRefresher you have to identify the data that can be used by the extract request. This data can come from a single source or several sources.

   **Note:** When extracting data from a MVS or VM relational source, you only need to complete the next two steps. There is no need to create the data descriptions for the source data, as DataRefresher uses the data descriptions which are stored in the database catalog.

2. Create the extract request, and store the extract in the Extract Library (EXTLIB).

   The DataRefresher extract request identifies:
   - Where the source data is stored
   - What data is to be extracted
   - Where the data is to be placed

3. Run the Extract Manager to process the extract request.

   The Data Extract Manager processes the extract request for non-relational databases, using the data description to identify the data to be extracted, and sends the extracted data to the target.

   Relational database extract requests are processed by the Relational Extract Manager, and do not require data descriptions as they use the information in the relational tables.

Figure 2 shows how the description of the data and the extract requests are used by the Data Extract Manager to select the source data and send it to the target.

![Diagram](image-url)

*Figure 2. How DataRefresher works*
Step 1. Describing the source data

When extracting data from a non-relational source, you must create a description of the data you want to extract. A data description describes the structure and organization of a data source to DataRefresher. The following are the data description types used by DataRefresher:

**DXTFILE description**

This describes a non-IMS file from which the DEM can extract data. These can be simple files (one record type) or structured files (multiple record types or files with internal segments).

An internal segment is a segment that is positioned within another segment, and contains a group of data that is repeated. The segment containing the internal segment is called a containing segment or a parent segment.

When you create your file description you have to:

- Assign names to the file description, the segments and the fields.
- Describe how the file is organized.
- Specify whether any exit routines are used when the data is accessed.

You must also decide whether you are going to create multiple file descriptions for the file. You can create several file descriptions for the same file, with each description including different fields or segments. If you have several files with the same format, a single description can be used to access all of the files.

**DXTPSB description**

This describes an IMS PSB through which the DEM (Data Extract Manager) can access one or more IMS databases. The DXTPSB is a collection of DXTPCBs, and corresponds to the IMS PSB.

When creating a DXTPSB description you must:

- Assign the following names in the description:
  
  - DXTPSB
  - DXTPCB
  - IMS fields
  - IMS segments

- Organize the DXTPSB and PCB to include all the fields and segments from which data is to be extracted. The segments you specify in the DXTPSB must reside in a logical path, and must have their root and parent segments included in the description.

**DXTVIEW description**

This defines a view of a DXTFILE or DXTPSB, and is used in an extract request to identify the data to be extracted. The DXTVIEW is like a window to the DXTFILE or DXTPSB description, through which segments and fields can be selected for extraction.

**User data type description**

This defines a user data type that is not supported by DataRefresher. The user data type description is used to transform the unsupported data type into a data type that is supported. A user data type description describes the format of the data and is used with User Data Type exit routines.
When the data descriptions have been created they must be stored in the File Description Table Library (FDTLIB). The DataRefresher User Input Manager (UIM) is provided in the MVS environment to validate the data descriptions and place them in the FDTLIB.

You can use any of the following facilities to create data descriptions in an MVS environment. In a VM environment, you can use the DataRefresher dialogs or the system editor.

**Structures Access Program (SAP)**
A program that uses user-specified data structures to generate data descriptions and extract request statements. For more information about SAP see Chapter 10, “Extracting data using the SAP” on page 89.

**Dictionary Access Program (DAP)**
A program for generating data descriptions of source files and non-relational databases from which data can be extracted. DAP retrieves these descriptions from the IBM OS/VS DB/DC Data Dictionary. For more information about DAP see Chapter 11, “Creating data descriptions using the DAP” on page 107.

**Online DataRefresher commands**
DataRefresher provides a series of commands to let you create data descriptions and extract requests online. For a list of these commands see “Online DataRefresher commands” on page 51, and for more detailed information see the *DataRefresher Command Reference*.

**DataRefresher dialogs**
DataRefresher provides a series of panels for generating data descriptions and extract requests.

For detailed information about the DataRefresher dialogs see Part 2, “Using the Administrative Dialogs” on page 57 and Part 3, “Using the End User Dialogs” on page 155.

**System editor**
You can use your system editor to create your data descriptions based on a series of models supplied with DataRefresher.

**Note:** All of the preceding ways of creating a data description involve the use of the CREATE command with one of the following statements:

- DATATYPE
- DXTFILE
- DXTPSB
- DXTVIEW

For more information about the CREATE command and these statements, see the *DataRefresher Command Reference*. 

Chapter 2. How DataRefresher Works 11
Step 2. Creating an extract request

An extract request tells Extract Manager to move specific data from one location in a data source place it in another location in the same database or another target database. To do this, the Extract Manager needs to know:

- Where the source data is
- What data you want extracted
- Where to put the extracted data

The extract request is created using the SUBMIT command with an EXTRACT statement. This command uses a subset of the SQL SELECT statement to build the extract request. SUBMIT identifies where the data to be extracted can be found, and where it is to be placed after it has been extracted.

You create the extract request with the SUBMIT command, using the EXTRACT statement, and place it in the extract request library (EXTLIB) from where it is processed by the Extract Manager.

You can use any of the following facilities to create extract requests in an MVS environment. In a VM environment, you can use the DataRefresher dialogs or the system editor.

**SAP (Structures Access Program)**
This creates an extract request based on stored information. For more information about using SAP to create an extract request see Chapter 10, "Extracting data using the SAP" on page 89.

**DataRefresher dialogs**
These creates an extract request using a series of ISPF panels.

For detailed information about the DataRefresher dialogs see Part 2, "Using the Administrative Dialogs" on page 57 and Part 3, "Using the End User Dialogs" on page 155.

**Online DataRefresher commands**
These provide a command for creating extract requests. This command, DSEND, is described in DataRefresher Command Reference.

**System editor**
You can use the editor to create a extract request and its related JCL without using the tools provided by DataRefresher based on the model supplied with DataRefresher.

When the extract request has been created it must be stored in the Extract Library (EXTLIB). The DataRefresher User Input Manager (UIM) is provided to validate non-relational extract requests and place them in the EXTLIB. Relational extract requests are executed when you submit them.
For every extract request in the Extract Library (EXTLIB), there must be a specific data description in the FDTLIB to describe the data to be extracted.

Step 3. Running the extract manager

When you create an extract request you have the option of specifying JCS to process the data after it has been extracted. This JCS must be specified when you create the extract request. For non-relational sources it gets stored by the UIM in the EXTLIB and is used by the DEM when the data is extracted. For relational sources the extract runs when it is submitted and then the JCS can be used to process the extracted data.

When you process an extract request, you can:

- Combine the extracted data with a load utility control deck, and job control statements to tell DataRefresher where to place the data.
- Write the extracted data to a data set or file.

There are two programs that can be used to process extract requests on the host:

Data Extract Manager (DEM)

Processes non-relational data extract requests previously sorted in the EXTLIB by the UIM.

The DEM is used to extract data from the non-relational sources supported by DataRefresher.

Relational Extract Manager (REM)

Processes relational data extract requests.

Because the data is identified in the database's catalog, relational extracts do not require DataRefresher data descriptions and the UIM is not used.
Chapter 3. How to create data descriptions

This section explains how data description commands are coded. You can use the following methods to create data descriptions:

- An editor
  You can code data descriptions with an editor without any DataRefresher tools to help you. See "System editor" on page 52.

- Administrative Dialogs
  See Chapter 5, "What are Administrative Dialogs?" on page 59.

- DataRefresher online commands
  See the DataRefresher Command Reference.

- The Dictionary Access Program (DAP)
  See Chapter 11, "Creating data descriptions using the DAP" on page 107.

- The Structures Access Program (SAP)
  See Chapter 10, "Extracting data using the SAP" on page 89.

To use any of the above methods, you have to know how to write DataRefresher commands that create the data descriptions. This section shows you how to write these commands. It does not explain the individual methods listed above.

Overview of tasks

1. Write data descriptions with the necessary CREATE (UIM) commands.

2. Submit the data description commands to the UIM.

   The UIM validates the data descriptions and places them in the FDTLIB.

   Use one or more of the following DataRefresher commands, as appropriate, to create data descriptions:

   - CREATE DATATYPE
   - CREATE DXTFILE
   - CREATE DXTPSB
   - CREATE DXTVIEW

   The commands you use depend on the source data. Table 1 on page 16 lists various examples of source data and the commands that you can utilize with them.
<table>
<thead>
<tr>
<th>Source data</th>
<th>CREATE DATATYPE</th>
<th>CREATE DXTFILE</th>
<th>CREATE DXTPSB</th>
<th>CREATE DXTVIEW</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS database</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>IMS database with a user data type exit</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Physical sequential or VSAM data set</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Physical sequential or VSAM data set with a user data type exit</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Local source accessed by a GDI exit</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Local source accessed by a GDI exit with a user data type exit</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>See page:</strong></td>
<td>16</td>
<td>18</td>
<td>24</td>
<td>39</td>
</tr>
</tbody>
</table>

**CREATE DATATYPE**

The CREATE DATATYPE command creates a new user-defined data type description that describes the format of user data type fields.

User data type exits are written to transform data types unsupported by DataRefresher into a type that is supported by DataRefresher. For the user data type exits to do this, you must create a user data type description that describes the format of the data. For more information about user data type exits, see *DataRefresher Exit Routines*.

To use a user data type description, you also write a DXTFILE description or a DXTPSB description (using CREATE DXTFILE or CREATE DXTPSB). A DXTFILE or DXTPSB description then refers to the user data type that has been described with the CREATE DATATYPE command.

**Specifying source and target data**

To create a user data type description, you must specify source (user format) and target (DataRefresher supported format) information. The following table shows the keywords to use in the CREATE DATATYPE command to describe the user data type:

<table>
<thead>
<tr>
<th>Describing the user data type</th>
<th>Describing the source data</th>
<th>Describing the target data</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE DATATYPE</td>
<td>SRCCTYPE=</td>
<td>TRGTYPE=</td>
</tr>
<tr>
<td>EXIT=</td>
<td>SRCBYTES=</td>
<td>TRGBYTES=</td>
</tr>
<tr>
<td>DESC=</td>
<td>SRCSCALE=</td>
<td>TRGSSCALE=</td>
</tr>
</tbody>
</table>

For a complete description of the CREATE DATATYPE keywords, see *DataRefresher Command Reference*.
Examples of the CREATE DATATYPE command

In the first example, a user data type is described and a DXTFILE is created using the user data type:

```
CREATE DATATYPE SRCTYPE=XX,
   EXIT=XXEXIT,
   SRCSOURCE=5,
   SRCSIZE=0,
   TRGTYPE=V,
   TRGBYTES=6,
   TRGSIZE=2,
   DESC='USER DATA TYPE IN XX FORMAT';

CREATE DXTFILE
   NAME=USINGXX, ACCESS=VK, DESC='FILE USING XX DATA TYPE'
SEGMENT NAME=STAFF, DESC='STAFF INFORMATION'
   FIELD NAME=FLD1, TYPE=C, BYTES=8, START=1
   FIELD NAME=FLD2, TYPE=XX,  1
      BYTES=5,
      START=9,
      SCALE=0  3
   FIELD NAME=FLD3, TYPE=H, BYTES=2, START=14;
```

1. This user data type is named XX. It is identified by the SRCTYPE keyword on the CREATE DATATYPE command and the TYPE keyword on the CREATE DXTFILE command.

2. The exit used to transform fields of type XX to a type supported by DataRefresher is named XXEXIT.

3. The SRCSOURCE keyword on the CREATE DATATYPE command specifies how many bytes the field contains before it is transformed to a DataRefresher supported type. In this case SRCSOURCE=5.

4. The BYTES keyword on the CREATE DXTFILE command specifies to DataRefresher the same information—the number of bytes of the source data before it is transformed. For this reason, the BYTES keyword can be omitted from the CREATE DXTFILE command, and the value from the SRCSOURCE keyword is assumed. If you do include BYTES, it must match the value of the SRCSOURCE keyword, or an error message is issued.

5. The SRCSIZE keyword on the CREATE DATATYPE command specifies the scale of the source data before it is transformed to a DataRefresher supported type. (The scale indicates where the decimal point is placed.)

6. The SCALE keyword on the CREATE DXTFILE command specifies to DataRefresher the same information—the scale of the source data before it is transformed. For this reason, the SCALE keyword can be omitted from the CREATE DXTFILE command, and the value from the SRCSIZE keyword is assumed. If you do include SCALE, it must match the value of the SRCSIZE keyword, or an error message is issued.

7. The TRGTYPE keyword specifies the supported DXT data type of the field after it is transformed by the exit.

8. The TRGBYTES keyword specifies the length, in bytes, of the field after it is transformed by the exit into a DataRefresher supported data type.

9. The TRGSIZE keyword specifies the scale of the field after it is transformed by the exit into a DataRefresher supported data type.
In the next example, another user data type description is created, followed by two DXTFILE descriptions. The VARIES keyword is used, which lets you write one user data type description and use it for more than one file or PSB description that contain user data type fields with varying lengths. Each file or PSB description specifies the length the CREATE DATATYPE command leaves out when using VARIES. The exit returns a value for the VARIES specified on TRGBYTES, and it validates the SRCBYTES value specified in the file description.

```
CREATE DATATYPE SRCTYPE=YY,
    EXIT=YYEXIT,
    SRCBYTES=VARIES, 1
    TRGTYPE=C,
    TRGBYTES=VARIES, 2
    DESC='USER DATA TYPE IN YY FORMAT';

CREATE DXTFILE
    NAME=USINGYY, ACCESS=VK, DESC='FILE USING YY DATA TYPE'
    SEGMENT NAME=DEPT, DESC='DEPT INFORMATION'
        FIELD NAME=FLD1, TYPE=C, BYTES=8, START=1
    FIELD NAME=FLD2, TYPE=YY, 3
        BYTES=5, START=9
    FIELD NAME=FLD3, TYPE=H, BYTES=2, START=14;

CREATE DXTFILE
    NAME=AGAINYY, ACCESS=VK, DESC='ANOTHER FILE USING YY DATA TYPE'
    SEGMENT NAME=REST, DESC='RESTAURANT INFORMATION'
        FIELD NAME=MENU, TYPE=C, BYTES=15, START=1
    FIELD NAME=ORDERS, TYPE=YY, 4
        BYTES=8, START=16
    FIELD NAME=HELP, TYPE=C, BYTES=10, START=25;
```

1. The length of the source field varies, specified with SRCBYTES=VARIES.
2. The length of the target field varies, specified with TRGBYTES=VARIES.
3. The BYTES keyword on the first CREATE DXTFILE command specifies the length of SRCBYTES (the length of the source field) as 5. The length of the target field (TRGBYTES) is returned by the exit at field definition time (when the CREATE DXTFILE command is executed).
4. The BYTES keyword on the second CREATE DXTFILE command specifies the length of SRCBYTES (the length of the source field) as 8. The length of the target field (TRGBYTES) is returned by the exit at field definition time (when the CREATE DXTFILE command is executed).

**CREATE DXTFILE**

The CREATE DXTFILE command describes the following non-IMS data sources:

- Physical sequential data sets
- VSAM data sets
- Sources accessed by a GDI exit

To extract data from non-IMS data sources, the DEM needs a description of the file. To describe your source data, you need to know the structure of that file. For details about file structures that DataRefresher recognizes, see the *DataRefresher Command Reference.*
The DXTFILE description, created with the CREATE DXTFILE command, describes to DataRefresher the structure and organization of the file. To create a DXTFILE description, the following steps are necessary:

- Assign names in a DXTFILE description
- Describe the organization of the file to DataRefresher
- Specify user-written exit routines

Assigning names in a DXTFILE description

When you create a DXTFILE description, you assign names to the file description, to the segments (if it is a structured file), and to the fields.

- The name you assign to the DataRefresher file description need not be the actual name of the file, but it must be distinct from the names of any other DXTFILE description in the FDTLIB.

- The name you assign to each segment in the DXTFILE description must be distinct from the names of any other segment in the description.

- The name you assign to each field in a simple DXTFILE description must be distinct from the names of any other field in the description. Also, the name you assign to each field within the same segment of a structured DXTFILE description must be distinct from the names of any other fields in that segment. (You can assign two fields in different segments the same name because the field descriptions in one segment are independent from those in other segments.)

The names you assign to DXTFILEs, segments, and fields can be DataRefresher names, DataRefresher quoted names, or DBCS names. For more information about names for DXTFILE descriptions, see DataRefresher Command Reference.

Note: If RACF is used to protect the DataRefresher data descriptions in the FDTLIB and extract requests in the EXTLIB, the name you specify must conform to the RACF naming conventions established for your site. For details on preparing to use RACF with DataRefresher, refer to DataRefresher Administration Guide.

Describing the organization of the file

The CREATE DXTFILE command performs three basic tasks:

- Describes the file (DXTFILE statement)
- Describes each segment included in your description (SEGMENT statement)
- Describes each field included in your description (FIELD statement)

You can describe any portion of the file that you want to include. If there is information in one segment that you don't need, don't include it in the CREATE DXTFILE command. If there are segments or fields that you don't want others to see (even though they have access to the FDTLIB that contains your data descriptions), you do not need to include them. You can write one CREATE DXTFILE command that describes an entire file, then write multiple views. For more information about restricting access to a DXTFILE, see "CREATE DXTVIEW" on page 39.

The characteristics of the file, the segments, and the fields are specified with the keywords and values on the CREATE DXTFILE command. The following table contains all keywords included on the CREATE DXTFILE command:
Table 2. Keywords in the CREATE command

<table>
<thead>
<tr>
<th>Describe the file (DXTFILE statement)</th>
<th>Describe a segment (SEGMENT statement)</th>
<th>Describe a field (FIELD statement)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE DXTFILE</td>
<td>SEGMENT</td>
<td>FIELD</td>
</tr>
<tr>
<td>NAME=</td>
<td>NAME=</td>
<td>NAME=</td>
</tr>
<tr>
<td>ACCESS=</td>
<td>FORMAT=</td>
<td>START=</td>
</tr>
<tr>
<td>DDNAME=</td>
<td>FREQ=</td>
<td>TYPE=</td>
</tr>
<tr>
<td>DATAEXIT(EXIT)=</td>
<td>OCCURS=</td>
<td>BYTES=</td>
</tr>
<tr>
<td>XBYTES=</td>
<td>START=</td>
<td>LFIELD=</td>
</tr>
<tr>
<td>EODCALL=</td>
<td>BYTES=</td>
<td>SCALE=</td>
</tr>
<tr>
<td>FREQ=</td>
<td>NEXT=</td>
<td>CONV=</td>
</tr>
<tr>
<td>GDIEXIT=</td>
<td>PARENT=</td>
<td>SEQFLD=</td>
</tr>
<tr>
<td>GDIXTYPE=</td>
<td>DESC=</td>
<td>SEQUENCE=</td>
</tr>
<tr>
<td>DETAIL=</td>
<td>UNIQUE=</td>
<td>DESC=</td>
</tr>
<tr>
<td>DESC=</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For the syntax diagram and the complete descriptions of each keyword and value, see DataRefresher Command Reference.

**Note:** DataRefresher supports both character and numeric key fields in VSAM files. Numeric key support includes packed decimal, halfword binary, fullword binary, and zoned decimal. Optimal numeric key performance is achieved when the "=" boolean operator is used in the extract request.

**Describing the file**

Give the name that you choose for your DXTFILE description to the person who writes the JCL to run the DEM. The DEM uses the name to process extract requests that refer to the DXTFILE description.

If your site uses RACF to protect its data items and your data descriptions need to be protected, consider giving this name to the RACF administrator.

**Using multiple file descriptions**

There can be multiple DXTFILE descriptions of the same non-IMS file in an FDTLIB. Each of these descriptions contains only the segments and fields you want to be included. Conversely, a single DXTFILE description can describe a number of different non-IMS files with the same format.

**Note:** When one DXTFILE description describes more than one non-IMS file, you can extract data from only one of the described files on any particular run of the DEM.

**Specifying user-written exit routines**

You can write an exit routine to help you perform specialized functions. With the CREATE DXTFILE command, you can specify the name and type of exit routine you want to use to access your data. The types of exit routines you can specify are data exits, date/time conversion exits, and generic data interface (GDI) exits. For a complete description of exit routines, see DataRefresher Exit Routines.

**Examples of the CREATE DXTFILE command**

The following examples show three data descriptions for three types of files. A file from the DataRefresher sample data is being described in each case.

**Note:** Not all keywords available with the CREATE DXTFILE command are used in these examples. For the syntax diagram and complete keyword descriptions, see the DataRefresher Command Reference.
Simple file (one record type)
You can write a data description of a file containing just the ORGSEG records in the PSEQDEPT file. (PSEQDEPT is not a simple file, but the ORGSEG segment alone resembles a simple file.)

The data is structured like this:

<table>
<thead>
<tr>
<th>DEPT</th>
<th>SUFFIX</th>
<th>DEPTNAME</th>
<th>MANAGER</th>
<th>DIVISION</th>
<th>LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEPT</td>
<td>SUFFIX</td>
<td>DEPTNAME</td>
<td>MANAGER</td>
<td>DIVISION</td>
<td>LOCATION</td>
</tr>
</tbody>
</table>

Figure 3. Simple file

The CREATE DXTFILE command would look like this:

```
CREATE DXTFILE NAME=PROJECTS, ACCESS=PS, FREQ=20,
   DESC='DEPARTMENT PROJECTS'
FIELD NAME= DEPT, START= 1, TYPE=H, BYTES= 2, DESC='DEPT NUMBER' A
FIELD NAME= SUFFIX, START= 3, TYPE=H, BYTES= 2, DESC='SUFFIX - 00'
FIELD NAME= DEPTNAME, START= 5, TYPE=C, BYTES=14, DESC='NAME OF DEPT'
FIELD NAME= MANAGER, START=19, TYPE=H, BYTES= 2, DESC='MGR. NUMBER'
FIELD NAME= DIVISION,START=21, TYPE=C, BYTES=10, DESC='DIVISION'
FIELD NAME= LOCATION,START=31, TYPE=C, BYTES=13, DESC='LOCATION';
```

Figure 4. DXTFILE data description of a simple file

1. PROJECTS is the name of the DXTFILE description.
2. ACCESS=PS indicates that PROJECTS describes a physical sequential data set.
3. FREQ=20 indicates the estimated number of records in this file. This keyword is optional and used for optimization purposes.
4. A FIELD statement describes each field. The name, starting location, type, and length are specified for each field in this example. The length of the field does not need to be specified if it is inherent in the data type. For example, an H-type field is always 2 bytes in length, therefore the length is inherent in the data type and need not be specified. There are no SEGMENT statements because this is a simple file.

Structured file (multiple record types, no internal segments)
You can write a data description of the VSAM file named VSAMDEPT in the sample data. There are two record types, ORGSEG and PROJSEG, but there are no internal segments.
The data is structured like this:

```
<table>
<thead>
<tr>
<th>PROJSEG</th>
<th>DEPTKEY</th>
<th>DEPTNUM</th>
<th>PROJNUM</th>
<th>STARTD</th>
<th>ENDD</th>
<th>TIMESTMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEPT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ORGSEG</td>
<td>DEPTKEY</td>
<td>DEPTNAME</td>
<td>MANAGER</td>
<td>DIVISION</td>
<td>LOCATION</td>
<td></td>
</tr>
<tr>
<td>DEPT</td>
<td>SUFFIX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Record 1 format

Record 2 format

Figure 5. Structured file with multiple record types

The CREATE DXTFILE command would look like this:

```
CREATE DXTFILE NAME=VSAMDEPT,
    ACCESS=VK,
    FREQ=20,
    DESC='PROJECT INFORMATION'
SEGMENT NAME=PROJSEG,
    DESC='SEGMENT FOR THE PROJ RECORDS'
    FIELD
        NAME= DEPT,START= 1, TYPE=H, SIZE= 2,
        DESC='DEPT NUMBER'
        FIELD NAME= PROJNUM, START= 3, TYPE=H, SIZE= 2,
        DESC='PRODUCT NO'
        FIELD NAME= PROJNUM, START= 5, TYPE=C, SIZE= 4,
        DESC='PROJECT NO'
        FIELD NAME= STARTD, START= 9, TYPE=A, SIZE=10,
        DESC='START DATE'
        FIELD NAME= ENDD, START=19, TYPE=A, SIZE=10,
        DESC='END DATE'
        FIELD NAME= TIMESTMP,START=29, TYPE=5, SIZE=26,
        DESC='TIME STAMP'
SEGMENT NAME=ORGSEG,
    DESC='SEGMENT FOR THE DEPT RECORDS'
    FIELD NAME= DEPT,START= 1, TYPE=H, SIZE= 2,
        DESC='DEPT NUMBER'
    FIELD NAME= SUFFIX, START= 3, TYPE=H, SIZE= 2,
        DESC='SUFFIX-00'
    FIELD NAME= DEPTNAME,START= 5, TYPE=C, SIZE=14,
        DESC='NAME OF DEPT'
    FIELD NAME= MANAGER, START=19, TYPE=H, SIZE= 2,
        DESC='MANGR. NUMBER'
    FIELD NAME= DIVISION,START=21, TYPE=C, SIZE=10,
        DESC='DIVISION'
    FIELD NAME= LOCATION,START=31, TYPE=C, SIZE=13,
        DESC='LOCATION';
```

Figure 6. DXTFILE description of a structured file

1. VSAMDEPT is the name of the DXTFILE description.
2. ACCESS=VK indicates that VSAMDEPT describes a VSAM KSDS file. (If you are describing a VSAM ESDS file, specify ACCESS=VE.)
3. FREQ=20 indicates that the estimated number of records in this file is 20. This is optional and used for optimization purposes.
4. A SEGMENT statement identifies one of the two record types. The first is called PROJSEG.
5. A FIELD statement identifies each field in the segments.
6. The name, starting location, type, and length are specified for each field in this example. The length of the field does not need to be specified if it is inherent in the data type. For example, an H-type field is always 2 bytes in length, therefore the length is inherent in the data type and need not be specified.
7. ORGSEG is the second segment.
Structured file (internal segment, one record type)

You can describe the physical sequential data set named PSEQSTAFF in the sample data. This data set is structured with one parent segment and one internal segment. In this example, there are no nested internal segments.

The structure of the data looks like this:

<table>
<thead>
<tr>
<th>STAFFSEG</th>
<th>HISTSEG</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>NAME</td>
</tr>
</tbody>
</table>

Figure 7. Structured file with internal segments

The CREATE DXTFILE command would look like this:

```
CREATE DXTFILE NAME=PSEQSTAFF, 
   ACCESS=PS, 
   FREQ=20, 
   DESC='EMPLOYEE DATA'
SEGMENT NAME=STAFFSEG, 
   DESC='EMPLOYEE STATISTICS'
   FIELD NAME= ID, START= 5, TYPE=H, BYTES= 2, DESC='EMP ID'
   FIELD NAME= NAME, START= 7, TYPE=C, BYTES= 9, DESC='EMP NAME'
   FIELD NAME= DEPT, START=16, TYPE=H, BYTES= 2, DESC='DEPT NUM'
   FIELD NAME= JOB, START=18, TYPE=C, BYTES= 5, DESC='JOB NAME'
   FIELD NAME= YEARS, START=23, TYPE=B, BYTES= 1, DESC='YRS WORKED'
   FIELD NAME= SALARY, START=24, TYPE=P, BYTES= 4, SCALE=2,
                          DESC='MONTHLY SALARY'
   FIELD NAME= COMM, START=28, TYPE=P, BYTES= 4, SCALE=2,
                       DESC='AVERAGE COMMISSION'
   FIELD NAME= NUM_HIST, START=32, TYPE=B, BYTES= 1,
                      DESC='NUMBER OF HIST SEGMENTS'
SEGMENT NAME=HISTSEG, PARENT=STAFFSEG, 
   FORMAT=FI,
   BYTES=24,
   OCCURS=NUM_HIST,
   START=NUM_HIST+1,
   DESC='SEGMENT FOR HISTORY RECORDS'
   FIELD NAME= DATE, START= 1, TYPE=A, BYTES=10,
                        DESC='CHANGE DATE'
   FIELD NAME= PAST_JOB, START=11, TYPE=C, BYTES= 5, DESC='JOB NAME'
   FIELD NAME= YRS_WKED, START=16, TYPE=B, BYTES= 1, DESC='YRS WORKED'
   FIELD NAME= SSAL, START=17, TYPE=P, BYTES= 4, SCALE=2,
                       DESC='STARTING SALARY'
   FIELD NAME= ESAL, START=21, TYPE=P, BYTES= 4, SCALE=2,
                       DESC='ENDING SALARY';
```

Figure 8. DXTFILE description of a structured file

1. PSEQSTAFF is the name of the DXTFILE description.
2. ACCESS=PS indicates that PSEQSTAFF describes a physical sequential data set.
3. FREQ=20 indicates that the estimated number of records in this file is 20. This is optional and used for optimization purposes.
4. and 5. There are two segments: STAFFSEG and HISTSEG. STAFFSEG is the parent segment of HISTSEG, specified: PARENT=STAFFSEG.
6. FORMAT=FI indicates that HISTSEG is a fixed internal segment.
7. BYTES=24 indicates the length of the internal segment.
OCCURS=NUM_HIST indicates how many times the HISTSEG internal segment occurs within a particular STAFFSEG occurrence. This can be a number, for example, OCCURS=5. Or, as in this example, you can specify a field that contains the number of occurrences.

START=NUM_HIST+1 specifies the starting position for the HISTSEG segment. In this example, it follows the last field in the STAFFSEG segment.

A FIELD statement identifies each field in the segment.

The name, starting location, type, and length are specified for each field in this example. The length of the field does not need to be specified if it is inherent in the data type.

Accessing a file with your GDI record exit
Assume you want to write a data description of the VSAM file VSAMDEPT in the sample data again. This time, however, you have a GDI record exit, named GDIVSAM, that you have written. (Assume you have the exit routine, though it is not provided with the sample data.) The GDI record exit called GDIVSAM describes the VSAMDEPT file; therefore, the segments and fields do not need to be described in the CREATE DXTFILE command. The exit returns this description to the UIM so that it can update the FDTLIB.

The CREATE DXTFILE command would look like this:

```
CREATE DXTFILE
   NAME=VSAMDEPT,
   DESC='DESCRIBING VSAM FILE WITH GDI EXIT',
   FREQ=20,
   ACCESS=GDI,
   GDIEXIT=GDIVSAM,
   GDIXTYPE=RECORD,
   DETAIL=EXIT;
```

Figure 9. DXTFILE description accessed by a GDI record exit

1. VSAMDEPT is the name of the DXTFILE description.
2. ACCESS=GDI indicates that VSAMDEPT accesses a GDI data source. The name of the data source that is accessed is specified in the GDI exit.
3. GDIVSAM is the name of the GDI exit.
4. DETAIL=EXIT indicates that the details of the file being described are in the exit named GDIVSAM.

CREATE DXTPSB

To extract data from an IMS database, the DEM needs a description of the database as it is described in the IMS PSB. You supply this information to the DEM by creating a description of the relevant PSB with a CREATE DXTPSB command. To create a DXTPSB description, you need to assign names in DXTPSB descriptions and describe the structure and organization of a PCB to DataRefresher.
Assigning names in your DXTPSB description

When creating a DXTPSB description, you assign names to the PSB description, the PCB description, the fields of interest, and the segments containing those fields.

- The name of the DXTPSB description must be distinct from the name of any other DXTPSB description in the host FDTLIB. This name need not be the same as the name of the IMS PSB being described.

- The names of the DXTPCB descriptions within the PSB must be unique. In IMS, the PCBs don’t have names. When defining PCBs in DataRefresher, you give them your own names and define them in the same sequence they are defined in IMS (without skipping any except the last ones if not used).

- The names of the DXTPCB segments must match the corresponding segments in the IMS PCB. The IMS names are found in the Database Description (DBD) statements that define the IMS PCB. These names can be up to 8 characters long.

- The names of internal segments do not need to match anything; they are only used to describe this construct to DataRefresher. These names can be up to 32 characters long. The names of both database segments and internal segments within a PCB must be unique among themselves.

- The names of the fields in a DXTPCB description must match the corresponding fields you want to include from the IMS PCB. The IMS field names can be found in the DBD statements used to define the IMS PCB.

Describing the organization of the DXTPSB and DXTPCB

The CREATE DXTPSB command describes the following:

- The PSB (DXTPSB statement)
- The PCB (DXTPCB statement)
- Each segment included in your description (SEGMENT statement)
- Each field to be included in your description (FIELD statement)

You can include any segments and fields when describing a PCB within a particular PSB. By withholding any, you can prevent access to data through your DXTPSB description.
The segments you include in your PSB description must reside in a logical path. You must include the root and every parent of each segment you want to include. For example, in the hierarchy of the IMS sample database called SAMPPSB2, you include the HIST segment, and you must include the STAFF segment (the parent of HIST) and the ORG segment (the parent of STAFF).

```
   ORG
     ▼
    /  ^  \
   PROJ  STAFF
        ▼    \
           ▼
      HIST
```

The figure below shows how segments in an IMS database get described to IMS and to DataRefresher. There is a filtering process that limits what segments are available at each step.

IMS database: ▶ IMSPCB: ▶ DXTPCB: ▶ DXTVIEW:

1. SEG A → SENSEG A → SEGMENT A → SEGMENT A
2. SEG B → SENSEG B → SEGMENT B → SEGMENT B
3. SEG C → SENSEG C → SEGMENT C
4. SEG D → SENSEG D
5. SEG E

1 The IMS database is described by an IMS PCB with SENSEG statements. In your DXTPSB description, you can describe several PCBs.
2 Each DXTPCB describes any portion of the segments included in the IMS PCB description. Therefore, in your DXTPCB description, you can only describe segments that are in the IMS PCB description.
3 The view describes some or all of the segments in your DXTPCB description. You can extract data only from what is included in your DXTVIEW.
Finding information about the PSBs and PCBs

The control statements that define the database and PSB to IMS are the definitive source for your information. The DBDGEN control statements define the database. The PSBGEN statements define the PSB. Be sure to distinguish the IMS statements from corresponding UIM statements. For example, a DBDGEN FIELD statement defines a field in a database to IMS; the UIM FIELD statement describes that field to DataRefresher.

You can get most of your information from the DBDGEN control statements:

- **SEGMENT** statements include the lengths, formats, names, and parents of segments.
- **FIELD** statements include the names, lengths, locations, data characteristics of fields, and whether they are key fields. If they are key fields, you can also find out if they possess the ‘unique’ attribute.
- **DBD** statements include the IMS DL/I access method (for example, HISAM), and determine whether a database is logical or physical.
- **XDFLD** statements include the names of index fields, their source and target segments, and the fields from which they are composed.

You also can obtain other information from the PSBGEN statements:

- **SENSEG** statements indicate which segments in a database are visible through a database PCB.
- **SENFLD** statements indicate which fields in a segment are visible and where they appear in storage within their segment.
- **PCB** statements indicate whether a PCB sees its database as a secondary data structure associated with a ‘PROCSEQ’ index field, and the name of the index if it does.

**Note:** Information about embedded internal segments is not available in IMS DBDGEN and PSBGEN definitions because these are DataRefresher constructs.

Refer to the **IMS/ESA Version 3 Utilities Reference and IMS/ESA Version 3 Database Administration Guide** for more information about these statements and their uses. Or you can ask your database administrator or your information center for information about the structuring of the data and its meaning.

If your site maintains an IBM OS/VS DB/DC Data Dictionary, you can find much of what you need in that, including information about fields not defined in the database definitions (DBDs). You can consult the Dictionary through the standard Dictionary facilities or through DataRefresher’s Dictionary Access Program (the DAP). The DAP furnishes the information and punches it in the form of UIM statements that can create your DXTPSB description. Use of this program is discussed in Chapter 11, “Creating data descriptions using the DAP” on page 107.
CREATE DXTPSB keywords

The following table shows the keywords associated with the four tasks that the CREATE DXTPSB command performs:

<table>
<thead>
<tr>
<th>Describes the PSB (DXTPSB statement)</th>
<th>Describes a PCB (DXTPCB statement)</th>
<th>Describes a segment (SEGMENT statement)</th>
<th>Describes a field (FIELD statement)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE DXTPSB</td>
<td>DXTPCB</td>
<td>SEGMENT</td>
<td>FIELD</td>
</tr>
<tr>
<td>NAME=</td>
<td>NAME=</td>
<td>NAME=</td>
<td>NAME=</td>
</tr>
<tr>
<td>DESC=</td>
<td>DBACCESS=</td>
<td>PARENT=</td>
<td>TYPE=</td>
</tr>
<tr>
<td></td>
<td>DESC=</td>
<td>FORMAT=</td>
<td>START=</td>
</tr>
<tr>
<td></td>
<td>DBNAME=</td>
<td>BYTES=</td>
<td>BYTES=</td>
</tr>
<tr>
<td></td>
<td></td>
<td>START=</td>
<td>LFIELD=</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NEXT=</td>
<td>SCALE=</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OCCURS=</td>
<td>CONV=</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DATABEXIT=</td>
<td>SQFLD=</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XBYTES=</td>
<td>SEQUENCE=</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FREQ=</td>
<td>UNIQUE=</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAXNBR=</td>
<td>DESC=</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DESC=</td>
<td></td>
</tr>
</tbody>
</table>

For syntax diagrams and complete descriptions of each keyword in the CREATE DXTPSB command, see DataRefresher Command Reference.

Describing the PSB

Give the name that you choose for your DXTPSB description to the person who writes the JCL to run the DEM. The DEM uses the name to process extract requests that refer to the DXTPSB description.

If your site uses RACF to protect its data items and your data descriptions need to be protected, consider giving this name to the RACF administrator.

Describing the PCB

You do not have to include descriptions of every PCB in a PSB. However, you must specify the PCBs in order, and you cannot leave gaps. In other words, if you write a description of one PCB, you must also include a description for each preceding database PCB in the PSB. The DEM finds the PCB through its order relative to the other PCBs.

**PSBs containing alternate PCBs:** If you create a description of an IMS PSB that includes alternate PCBs, include a DXTPCB description for each alternate IMS PCB, placing these dummy DXTPCB descriptions first. This prevents the possibility that the DEM (running as a batch message processor) might use an IMS PSB containing alternate PCBs (TYPE=TP), which always precede the database PCBs in the PSB. Include a SEGMENT keyword for its root segment and follow that with a FIELD keyword for the root segment's sequence field. (Although these are dummy descriptions, they should be correct. The UIM has no way of distinguishing dummy descriptions from descriptions of valid database PCBs.)

**PCB access:** The High Speed Sequential Retrieval (HSSR) program speeds processing time in accessing IMS databases. The HSSR is a part of the IMS Data Base Tools (DBT) program product.
To use the HSSR program, specify DBACCESS=HSSR on each DXTPCB keyword. If you specify one DXTPCB in a DXTPSB with DBACCESS=HSSR, all other DXTPCBs within that DXTPSB must be specified with DBACCESS=HSSR. Each DXTPCB can specify only the segments in one hierarchical path which is your extract target. The corresponding IMS PCB must also contain one hierarchical path with segments that match the DXTPCB segments. The coding of the IMS PCB must adhere to HSSR PCB conventions. See the appropriate HSSR documentation for more information.

For more information about the DBACCESS keyword, see DataRefresher Command Reference.

**Describing the segment**
Describe both database and internal segments you want to include.

For a main storage database (MSDB) PCB you can write one segment and multiple internal segments. For a data entry database (DEDB) PCB you can write up to 127 database segments. For all other PCBs, you can write up to 255 segments.

Segments must be specified in hierarchic order (from top to bottom and left to right).

**Describing the field**
You can include only the fields you want accessed when creating a PSB description. Because the fields not described cannot be accessed through any view based on your description of the PCB, you can control what others can see through such DataRefresher views. even if these views are created by someone other than you.

*Field types:* There are two types of fields in DXTPSB descriptions:

- Ordinary fields
- Sequence fields

*Ordinary fields:* Ordinary fields are fields that you do not want identified as key or index fields. They can be key or index fields, but you choose not to classify them as either. For example, if the target segment for a particular index field contains a key field, you can describe the index field as a sequence field, and describe the key field as an ordinary field.

Observe these guidelines when describing ordinary fields:

- Do not use the SEQFLD, SEQUENCE, or UNIQUE keywords.
- Your DXTPCB description should describe ordinary fields as they look after processing by any data exit associated with the description.
- A PCB description does not need to be defined in the DBD for its database. For example, here is the definition of the HIST segment (from the DBD for the sample database SAMPPSB1):

```
SEGMENT NAME=HIST, PARENT=STAFF, BYTES=24, FREQ=1
FIELD NAME=PASTJOB, START=1, TYPE=C, BYTES=24
```

*Figure 10. Definition of the HIST segment*
In the CREATE DXTPSB command that describes SAMPPSB1 to DataRefresher, here is the definition of the HIST segment:

```
SEGMENT NAME=HIST,PARENT=STAFF,BYTES=24,FREQ=1
FIELD NAME=DATE,START=1,TYPE=A,BYTES=10
FIELD NAME=PASTJOB,START=11,TYPE=C,BYTES=5
FIELD NAME=YRSWKED,START=16,TYPE=B,BYTES=1
FIELD NAME=SSAL,START=17,TYPE=P,BYTES=4
FIELD NAME=ESAL,START=21,TYPE=P,BYTES=4;
```

**Figure 11. Definition of the HIST segment**

The length of the PASTJOB field in the DBD is 24 bytes. The length of all the fields in the HIST segment of the PSB description is also 24, but the layout has been divided into more fields. DataRefresher accepts any ordinary field descriptions as long as they are within the given segment.

**Sequence fields:** Sequence fields are either key or index fields. Observe these guidelines when describing a sequence field:

- Use a SEQFLD keyword. The SEQUENCE and UNIQUE keywords are optional.
- Do not use the LFIELD keyword.
- Identifying a sequence field can improve the performance of certain extract requests that use DXTPCB description. This is because the DEM can refer to sequence fields in its IMS DL/I calls when it executes a request that refers to the identified fields. Overall, such calls are handled more efficiently than calls that do not reference them. This is especially true when the identified fields are index or key fields for the root segments of HISAM, HDAM, or HIDAM databases.
- Assign a sequence field its DBD name for DataRefresher to use in IMS DL/I requests.
- If you use IMS secondary indexes, a key field is described in the segment that contains the field. An index field is described in the target segment, not the source segment (unless the two are the same).
- There can only be one sequence field in each segment.
- When using a data exit to preprocess your source data, describe the field as it looks in storage (not as it looks after processing by the exit) so that DataRefresher can implement its search strategy. If your exit attempts to change the length or contents of a sequence field, DataRefresher suspends execution of the current batch of extract requests and requeues it for later processing.
- You can identify only one index field in a DXTPCB description. The index field identified must be named with PROCSEQ in the PSBGEN PCB statement of the PCB definition. This index field causes the PCB to see a secondary data structure that may or may not be the hierarchic structure of the database for the PCB. In this secondary structure, the root segments are the target segments for the index field, and these root segments are sequenced by the index field.
- You must identify a ‘PROCSEQ’ index field, even if the target segment contains a key field. Identifying the key field implies that the key field, rather
than the index field, sequences the root segments of the secondary structure (that is, the target segments of the index). This misinformation can lead to severely degraded performance when your DXTPSB description is used to extract data.

- If you use the DAP to create a DXTPSB description, the field descriptions that it generates (for the root segment in a PCB with an index field) may need to be edited. For more information see Chapter 11, “Creating data descriptions using the DAP” on page 107.

**Note:** DataRefresher supports both character and numeric key fields in IMS files. Numeric key support includes packed decimal, halfword binary, fullword binary, and zoned decimal. Optimal numeric key performance is achieved when the “=” boolean operator is used in the extract request.

**Field lengths:** There are also two types of fields:

- **Fixed length**
  
  Fixed-length fields are always the same length. The length of a fixed-length field is specified with the BYTES keyword.

- **Variable length**
  
  Variable-length fields may be either of these data types:
  - variable-length character (VC)
  - variable-length graphic (VG)

  The length of a variable-length field must be specified in the length field, specified by the LFIELD keyword. The length field must precede the variable-length field. If the variable-length field is in a fixed length segment, and you do not specify a length field, DataRefresher assumes that the field length is whatever is left in the segment.

  Variable-length fields may not be defined to DataRefresher as a sort field or a sequence field.

  **Note:** If you are describing a user defined data type that has a variable length type (VC or VG) as its target type, no length field can be specified. The exit must return the length of the data to DataRefresher.

**Examples of the CREATE DXTPSB command**

The following examples show data descriptions for two sample databases. The first describes the database named SAMPPSB1. The second describes the database named SAMPPSB2.

**Note:** All of the keywords available with the CREATE DXTPSB command are not used in these examples. For the syntax diagram and complete keyword descriptions, see *DataRefresher Command Reference*.

**HSAM database without internal segments**

The organization of SAMPPCB1 (in SAMPPSB1) looks like this:
Figure 12 contains a portion of the SAMPPCB1 data description.

```
CREATE DXTPSB NAME=SAMPPSB1,DESC='PSB OVER THE HSAM DBASE'
    DXTPCB NAME=SAMPPCB1, 2
    DBACCESS=HSAM,
    DESC='ONE PCB OVER THIS HSAM DBASE'
SEGMENT NAME=ORG,  4
    PARENT=0,
    5
    BYTES=41,
    6
    DESC='ORGANIZATIONAL DATA'
FIELD NAME=DEPT,  7
    START=1,
    8
    TYPE=H,
    9
    BYTES=2,
    10
    DESC='DEPT NUMBER'
FIELD NAME=DEPTNAME, 11
    START=3,
    12
    TYPE=C,
    13
    BYTES=14,
    14
    SEQFLD=R,
    15
    DESC='PRODUCT NO'

... ... ...
SEGMENT NAME=PROJ, 13
    PARENT=ORG,
    14
    BYTES=52,
    15
    FREQ=4,
    DESC='CURRENT BUSINESS PROJECTS'
FIELD NAME=PROJNUM, 16
    START=3,
    17
    TYPE=C,
    18
    BYTES=4,
    19
    SEQFLD=R,
    DESC='PROJECT NO'

... ... ...
SEGMENT NAME=STAFF, 18
    PARENT=ORG,
    19
    BYTES=28,
    FREQ=10,
    DESC='EMPLOYEE STATISTICS'
...
FIELD NAME=SALARY, 20
    START=28,
    21
    TYPE=P,
    22
    BYTES=4,
    23
    SCALE=2,
    24
    DESC='YEARLY SALARY'
...
SEGMENT NAME=HIST, 23
    PARENT=STAFF,
    24
    BYTES=24,
    FREQ=1,
    DESC='EMPLOYMENT HISTORY'
... ... ...
```

Figure 12. DXTPSB description for an IMS database
1. SAMPPSB1 is the name of the DXTPSB.
2. SAMPPCB1 is the name of the DXTPCB.
3. DBACCESS=HSAM indicates that SAMPPCB1 is a Hierarchic Sequential Access Method (HSAM) database.
4. ORG is the first segment in this PCB.
5. PARENT=∅ indicates that ORG has no parent. ORG is the root segment.
6. BYTES=41 indicates that the ORG segment is 41 bytes long.
7. DEPT is the first field in the ORG segment.
8. START=1 indicates that DEPT starts at the first byte of the segment.
9. TYPE=H indicates that DEPT is an integer represented in a halfword.
10. BYTES=2 indicates that this field is two bytes long (the length of a halfword). Because the type of this field is H, the length is automatically determined to be two bytes long. Therefore, the keyword BYTES does not need to be specified. If you include the BYTES keyword and a TYPE that has an inherent length, the value you specify with BYTES must match the length determined by the type.
11. DEPTNAME is the next field in the ORG segment.
12. SEQFLD=R indicates that DEPTNAME is the sequence field of the ORG segment. DEPTNAME is a key or index field whose source and target segments are the same. There can only be one sequence field in each segment.

**Note:** The reference to source and target segments with sequence fields is actually IMS secondary index terminology. If you are not familiar with IMS secondary indexes, do not use the SEQFLD keyword. There can be only one sequence field.

13. PROJ is the second segment.
14. PARENT=ORG indicates that the PROJ segment is a child of the ORG segment.

**Note:** Because no format is specified for this segment, the value of the FORMAT keyword defaults to F, indicating a fixed-length database segment—it is not an internal segment.
15. FREQ=4 optimizes access strategies to the database by giving DataRefresher an estimate of the average number of occurrences of a segment within its parent.
16. PROJNUM is one of the fields in the PROJ segment.
17. SEQFLD=R indicates that PROJNUM is the sequence field of the PROJ segment. Because SEQFLD=R, PROJNUM is a key or index field whose source and target segments are the same. There can only be one sequence field in each segment.

**Note:** The reference to source and target segments with sequence fields is actually IMS secondary index terminology. If you are not familiar with IMS secondary indexes, do not use the SEQFLD keyword. There can be only one sequence field.
18. STAFF is the third segment.
PARENT=ORG indicates that STAFF is another child of the ORG segment.

SALARY is a field in the STAFF segment.

TYPE=P indicates that SALARY is a packed decimal field.

SCALE=2 indicates that the scale of the SALARY field is 2. A scale of 2 on a type P field gives this field a format for dollars and cents (to represent salary). The SCALE keyword specifies how many decimal digits are to the right of the decimal point.

HIST is the last segment.

PARENT=STAFF indicates that HIST is the child (not an internal segment) of STAFF.
**HISAM database with internal segments**

This example describes the SAMPPSB2 database, which is an HISAM database that contains internal segments.

![Diagram](image)

Organization of SAMPPCB2 database

- **root** (parent of PROJ and STAFF)
- child of ORG → PROJ
- child of ORG → STAFF
  - STAFF is the child of ORG and the containing segment of HIST
  - HIST is the internal segment of STAFF

Figure 13 on page 36 contains a portion of the SAMPCB2 data description.
Figure 13. DXTPSB description of an IMS database

1. SAMPPSB2 is the name of the DXTPSB.
2. SAMPPCB2 is the name of the DXTPCB.
3. DBACCESS=HISAM indicates that SAMPPCB2 is a Hierarchic Indexed Sequential Access Method (HISAM) database.
ORG is the first segment in the PCB.

PARENT=0 indicates that the ORG segment has no parent. ORG is the root segment.

BYTES=41 indicates that the ORG segment is 41 bytes long.

DEPT is the first field in the ORG segment.

START=1 indicates that DEPT starts at the first byte of the segment.

TYPE=H indicates that DEPT is an integer represented in a halfword.

BYTES=2 indicates that DEPT is two bytes long (the length of a halfword). Because the type of this field is H, the length is automatically determined to be two bytes long. Therefore, the keyword BYTES does not need to be specified. If you include the BYTES keyword and a TYPE that has an inherent length, the value you specify with BYTES must match the length determined by the type.

DEPTNAME is the next field in the ORG segment.

SEQFLD=R indicates that DEPTNAME is the sequence field of the ORG segment. DEPTNAME is a key or index field whose source and target segments are the same. There can only be one sequence field in each segment.

Note: The reference to source and target segments with sequence fields is actually IMS secondary index terminology. If you are not familiar with IMS secondary indexes, do not use the SEQFLD keyword in this way.

PROJ is the second segment.

PARENT=ORG indicates that the PROJ segment is a child of the ORG segment.

Note: Because no format is specified for this segment, the value of the FORMAT keyword defaults to F, indicating a fixed length database segment—it is not an internal segment.

FREQ=4 optimizes access strategies to the database by giving DataRefresher an estimate of the average number of occurrences of a segment within its parent.

PROJNUM is one of the fields in the PROJ segment.

SEQFLD=R indicates that PROJNUM is the sequence field of the PROJ segment. There can only be one sequence field in each segment.

Note: The reference to source and target segments with sequence fields is actually IMS secondary index terminology. If you are not familiar with IMS secondary indexes, do not use the SEQFLD keyword in this way.

STAFF is the third segment.

PARENT=ORG indicates that STAFF is another child of the ORG segment.

TYPE=P means that SALARY is a packed decimal field.

SCALE=2 specifies the scale of the SALARY field. A scale of 2 on a type P field gives this field a format for dollars and cents (to represent a salary). The SCALE keyword specifies how many decimal digits are to the right of the decimal point.

NUM_HIST is another field in the STAFF segment. This field contains the number of internal HIST segments that occur within the STAFF segment.
HIST is the name of another segment.

FORMAT=FI indicates that HIST is a fixed length internal segment.

PARENT=STAFF indicates that HIST is within the STAFF segment. STAFF is therefore the containing segment of HIST.

START=NUM_HIST+1 indicates that the starting place of the HIST segment always follows the NUM_HIST field.

FREQ=1 optimizes access strategies to the database by giving DataRefresher an estimate of the average number of occurrences of a segment within its parent.

OCCURS=NUM_HIST specifies exactly how many times the HIST segment occurs within the STAFF segment.

Deciding whether or not PSBs and PCBs can be used by DataRefresher

The following questions help determine when PSBs and PCBs can be used by DataRefresher:

For the PSB:

- Has the PSB been defined to IMS through the PSBGEN utility?
  
  You can create data descriptions for a nonexistent PSB, but you cannot use a DXTVIEW in an extract request of that PSB until the PSB is present in the IMS PSB library.

- Was a permanent application control block (ACB) generated for the PSB through the ACBGEN utility?
  
  You need a permanent ACB if you intend to use it for extractions in a batch message processing environment. See IMS/ESA Version 3 Database Administration Guide for more information.

- Did CMPAT=YES appear in the PSBGEN statement for the PSB generation?
  
  The compatibility feature ensures that DataRefresher sees the same PSB in a batch environment as it does in a batch message processing environment. When you specify CMPAT=YES in the PSBGEN statement, the I/O PCB that results is known to DataRefresher. You need not, therefore, write a dummy DXTPCB keyword when you write the command to create a DXTPSB description in DataRefresher.

- Was the PSB generated for either COBOL or ASSEMBLER applications?
  
  DataRefresher can handle only PSBs written for these two application types. Application type is indicated by the value of the LANG operand of the defining PSBGEN statement:

  | LANG=COBOL | for COBOL applications          |
  | LANG=ASSEM | for Assembler applications      |

  Figure 14. LANG operand of the defining PSBGEN statement

  Any value other than COBOL or ASSEM renders the PSB unusable for DataRefresher.

MVS and VM User's Guide
For individual PCBs:

- Is POS=M specified for the PCB?
  This must be specified in the PCB in order for DataRefresher to correctly extract data from more than one hierarchic path of the PCB. If it is not specified, incorrect output or a loop may result.

- Is the PCB a database PCB?
  A database PCB is one for which TYPE=DB appears in the PCB statement that was used in generating the PCB. DataRefresher can only process database PCBs. For any alternate PCB, which always appears before the database PCB, you must describe a dummy database PCB, as previously described. GSAM PCBs, which always follow the database PCBs, should be ignored in creating a DXTPSB description.

- Do the processing options include G and P?
  The processing options are specified with PROCOPT in the defining PCB statement. The options R, D, A, and G all imply the option G. The G and P options together enable the DEM to retrieve data through the PCB by using path calls. If these two options are not in effect, a data extraction that uses the PCB may fail.

  You may want to use the processing option O in conjunction with G to omit locking and thereby speed up the retrieval of data. Note, however, that when locking is not in effect during processing, the data that you extract may be inconsistent.

- Is there a SENSEG statement in the PCB definition for every segment from which you want to extract data?
  Without a SENSEG statement for a given segment, no data can be extracted from the segment.

- Were SENFLD statements used in defining the PCB?
  Field level sensitivity is in effect for any segment whose SENSEG statement is followed by one or more SENFLD statements. Field level sensitivity can greatly affect not only what fields can be seen, but also the in-storage appearance of the segment. For more information on the effects of field level sensitivity see \textit{DataRefresher Command Reference}, specifically the discussions of the CREATE DXTPSB command BYTES and FORMAT keywords associated with the SEGMENT keyword and the START keyword for an ordinary field.

---

**CREATE DXTVIEW**

After you write a CREATE DXTFILE command (for a nonrelational data source) or a CREATE DXTPSB command (for an IMS data source), you must write a DXTVIEW description. The view description, written with the CREATE DXTVIEW command, is like a window to the file or PSB descriptions. Various fields in a file or database can be seen through the view. The fields you include in the view description are the fields from which data can be extracted. After you specify a set of fields in a DXTVIEW, you name this view in the extract request as the source for extraction.

For example, suppose you want to extract data from a physical sequential data set, FILEX. You name this view VIEWX. You would then write a request to extract
data from FILEX, naming VIEWX as the data source. You can also write an extract request to extract data from multiple views.

To create a DXTVIEW description, you need to assign names and types in DXTVIEW descriptions.

Assigning names in DXTVIEW descriptions
The name must be distinct from the names of all the other DXTVIEW descriptions in the FDTLIB but it can be the same as the name of a DXTFILE description.

You can also assign an alias to any field in the DXTVIEW. Aliases are alternate names given to fields that may have the same names. For example, you can have a field named DEPT in the segment named SEGA, and you can have a field named DEPT in the segment named SEGB. You would assign an alias to at least one of these fields so DataRefresher can distinguish them. That alias must then be used instead of the field name when the field is referred to in an EXTRACT command.

The names you assign to your DXTVIEWs and field aliases can be up to 32 characters long. The types of names you can use are DataRefresher names, DataRefresher quoted names, and DBCS names. See DataRefresher Command Reference for more information about naming conventions.

Note: If RACF is used to protect the DataRefresher data descriptions in the FDTLIB and EXTLIB, the name you specify must conform to the RACF naming conventions established for your site. For details on setting up to use RACF with DataRefresher, refer to DataRefresher Administration Guide.

Assigning types of DXTVIEW descriptions
The types of DXTVIEW descriptions are:

- DXTVIEWs of DXTFILEs, including:
  - Simple files (one record type)
  - Structured files (multiple record types and/or internal segments)
- DXTVIEWs of DXTPSBs

Note: If you use a Generic Data Interface (GDI) select exit, you cannot create a DXTVIEW; DataRefresher automatically generates ‘dummy’ views (no fields defined) for GDI select files.

The following table contains the keywords of the CREATE DXTVIEW command:

<table>
<thead>
<tr>
<th>Naming the view (DXTVIEW statement)</th>
<th>Naming the DXTFILE (DXTFILE statement)</th>
<th>Naming the DXTPSB (DXTPSB statement)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE DXTVIEW</td>
<td>DXTFILE=</td>
<td>DXTPSB=</td>
</tr>
<tr>
<td>NAME=</td>
<td>SEGMENT=</td>
<td>DXTPCB=</td>
</tr>
<tr>
<td>DESC=</td>
<td>MINSEGEM=</td>
<td>SEGMENT=</td>
</tr>
<tr>
<td></td>
<td>FIELD(S)=</td>
<td>MINSEGEM=</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FIELD(S)=</td>
</tr>
</tbody>
</table>
Examples of CREATE DXTVIEW

The following examples show DXTVIEW descriptions of the DXTFILEs and DXTPSBs that were described in the previous sections.

**Note:** All keywords available with the CREATE DXTVIEW command are not used in these examples. For the syntax diagram and complete keyword descriptions, see *DataRefresher Command Reference*.

**DXTVIEW of a simple file**

Before you can create a DXTVIEW description of a simple file, your target FDTLIB must already contain a DXTFILE description of the file. You can create any number of DXTVIEW descriptions of a single DXTFILE description.

Because simple files contain only one record type, no segments exist in your DXTVIEW (or in the DXTFILE description). Your DXTVIEW should include any fields that you want to qualify for extraction.

This example uses the DXTFILE description of a simple file in the sample data. The name of this file description is PROJECTS. The structure and description are on page 21.

```
CREATE DXTVIEW NAME=PROJVIEW, 1
    DXTFILE=PROJECTS, 2
    FIELDS=*, 3
    DESC='DEPARTMENT ORGANIZATION DATA';
```

*Figure 15. DXTVIEW description of a simple file*

1 PROJVIEW is the name of the view.

2 PROJECTS is the name of the DXTFILE description that describes this data.

3 `FIELDS=*` indicates that all fields are included. The view of the sample data includes all the fields so you can extract from all the fields.

Example that limits these fields:

```
CREATE DXTVIEW NAME=PROJVIEW,
    DXTFILE=PROJECTS,
    FIELDS=(DEPTNAME, MANAGER, LOCATION), 1
    DESC='DEPARTMENT ORGANIZATION DATA';
```

1 You can extract data from only three fields: DEPTNAME, MANAGER, and LOCATION. The fields SUFFIX, DIVISION, and DEPT are not included.

**DXTVIEW of a structured file (multiple record types, no internal segments)**

If your DXTFILE description is of a structured file without internal segments, each DXTVIEW is of one segment only. Indicate the name of the DXTFILE description and the segment that your DXTVIEW defines, and list the names of the fields you want to be eligible for extraction. (You must include at least one field.) The fields you name in the DXTVIEW must be fields in the designated segment; they must be described in the DXTFILE description. Use the names of the fields that were used in the DXTFILE description, or use aliases that you define.
Before you can create the view for this structured file, a DXTFILE description of the data must be in the FDTLIB. You can create any number of DXTVIEW descriptions of a single DXTFILE description.

This example uses the DXTFILE description of a structured file without internal segments that exists in the sample data. The structure and description of this file are on page 22. The file description is named PSEQDEPT. There are two segments in this file, PROJSEG and ORGSEG. Because you can only use one segment in each DXTVIEW, this example shows two view descriptions, one using PROJSEG and one using ORGSEG.

```
CREATE DXTVIEW NAME=VSAMPROJECT,
    DXTFILE=VSAMDEPT,
    SEGMENT=PROJSEG,
    FIELDS=*,
    DESC='CURRENT BUSINESS PROJECTS';

CREATE DXTVIEW NAME=VSAMORG,
    DXTFILE=VSAMDEPT,
    SEGMENT=ORGSEG,
    FIELDS=*,
    DESC='DEPARTMENT ORGANIZATION DATA';
```

Figure 16. DXTVIEW description of a structured file

1 VSAMPROJECT is the name of the first view.
2 VSAMDEPT is the name of the DXTFILE description of this data.
3 PROJSEG is the segment that VSAMPROJECT includes.
4 FIELDS= * Indicates that all fields are selected.
5 VSAMORG is the name of the second view.
6 VSAMDEPT is the name of the DXTFILE description of this data.
7 ORGSEG is the segment that VSAMORG includes.
8 FIELDS= * indicates that all fields are selected.

All the fields are selected in the two views above so that you can extract from any field in those particular segments. These views describe the sample data.

The following example selects only certain fields from these views and defines some aliases:

```
CREATE DXTVIEW NAME=VSAMPROJECT,
    DXTFILE=VSAMDEPT,
    SEGMENT=PROJSEG,
    FIELDS=(PROJDEPT,PROJSEG.DEPT,PRODNUM,PROJNUM),
    DESC='CURRENT BUSINESS PROJECTS';

CREATE DXTVIEW NAME=VSAMORG,
    DXTFILE=VSAMDEPT,
    SEGMENT=ORGSEG,
    FIELDS=(ORGDEPT,ORGSEG.DEPT,DEPTNAME,MANAGER),
    DESC='DEPARTMENT ORGANIZATION DATA';
```

Figure 17. DXTVIEW description of a structured file

1 The fields included in the PSEQPROJ view are DEPT, PRODNUM, and PROJNUM. An alias is assigned to DEPT, called PROJDEPT. The other three
fields in the DXTFILE description, STARTD, ENDD, and TIMESTAMP, are not included; therefore, data in those fields cannot be extracted.

The fields included in the PSEQORG view are DEPT, DEPTNAME, and MANAGER. An alias is assigned to DEPT, called ORGDEPT. The other three fields in the DXTFILE description, SUFFIX, DIVISION, and LOCATION, are not included; therefore, data in those fields cannot be extracted.

Note: The fields named DEPT in both the ORGSEG and PROJSEG segments are assigned aliases. They do not need to be assigned aliases because they are specified in different views, but this example shows how it can be done.

DXTVIEW of a structured file (one record type, one internal segment)

If you create a DXTVIEW of a structured file that has internal segments, you need to specify the path you are viewing by specifying the bottom segment. This is because internal segments in a file impose a hierarchy like that of an IMS database. When you specify the bottom segment, all segments in the path from the top level segment (root) to the specified bottom segment are automatically included in the DXTVIEW.

After you specify the bottom segment, you can also specify the minimum segment. (This option is only available when creating a DXTVIEW of a structured file with internal segments.) The minimum segment indicates the lowest internal segment in the file that must be present for the data to qualify for extraction.

If a minimum segment is not specified with the MINSEGEM keyword, the segment you specified with the SEGMENT keyword is the minimum segment.

Consider data that is structured in the following way:

```
SEGA
  | SEGB
  |   | SEGC
  |   |   | SEGD
```

SEGA is the root segment.
SEGB is internal to SEGA.
SEGC is internal to SEGB.
SEGD is internal to SEGC.

If you specify SEGD as the bottom segment in the path with SEGMENT=SEGD, the path from A to D is included in the view. If you further specify that the minimum segment is SEGB with MINSEGEM=SEGB, all rows of data that have at least SEGB can qualify for extraction.

For example, assume these data records have the following structure:

```
<table>
<thead>
<tr>
<th>SEGA</th>
<th>SEGB</th>
<th>SEGC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEGA</td>
<td>SEGB</td>
<td>SEGC</td>
</tr>
<tr>
<td>SEGA</td>
<td>SEGB</td>
<td>SEGC</td>
</tr>
<tr>
<td>SEGA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEGA</td>
<td>SEGB</td>
<td></td>
</tr>
</tbody>
</table>
```
Only the first, second, and fourth data record would qualify for extraction in the example above. The third record does not have a SEGB. Because you specified SEGB as the minimum segment, data records that do not contain at least SEGB do not qualify for extraction.

Before you can create a view for this structured file, a DXTFILE description of the data must be in the FDTLIB. You can create any number of DXTVIEW descriptions of a single DXTFILE description. This example will use the DXTFILE description of a structured file with internal segments that exists in the sample data. The structure and description of this file are on page 23.

The file description is named PSEQSTAFF. There are two segments, STAFFSEG and HISTSEG. HISTSEG is an internal segment within STAFFSEG. The view description is called PSEQSTAFF (the view is provided with the DataRefresher sample data).

```
CREATE DXTVIEW NAME=PSEQSTAFF,
DXTFILE=PSEQSTAFF,
SEGMENT=HISTSEG,
FIELDS=*,
MINSEG=STAFFSEG, DESC='EMPLOYEE STATISTICS AND HISTORY';
```

1. PSEQSTAFF is the name of the DXTVIEW description.
2. PSEQSTAFF is the name of the DXTFILE description. The view and file descriptions can have the same names because they are different types of data descriptions.
3. HISTSEG is the bottom segment of the path. This means that both STAFFSEG and HISTSEG are included in this view and data from these segments can be extracted.
4. FIELDS=* indicates that any field can be extracted.
5. STAFFSEG is the minimum segment specified. This means that data records having at least the STAFFSEG segment can be extracted. Data records do not need to contain the HISTSEG segment to be extracted.

Note: To select only certain fields in the view, you could do the following:

```
CREATE DXTVIEW NAME=PSEQSTAFF,
DXTFILE=PSEQSTAFF,
SEGMENT=HISTSEG,
FIELDS=(NAME,DEPT,JOB,YRSWKED),
MINSEG=STAFFSEG, DESC='EMPLOYEE STATISTICS AND HISTORY';
```

1. The fields included in this view are NAME, DEPT, JOB, and YRSWKED. You may want to hide fields that contain information such as salary. By specifying only certain fields in the view, the data in all other fields does not qualify for extraction.
DXTVIEWs of IMS databases
Before you can create a DXTVIEW of an IMS database, a DXTPSB description must exist in your FDTLIB. You define your view based on the data description.

Within the DXTPSB description is a description of the PCB to be used when the DEM extracts data through your DXTVIEW. You can define any number of DXTVIEWs of a single DXTPCB.

Note: Use of the HSSR to access IMS data limits you to one DXTVIEW, because a DXTPCB can only contain the segments in one hierarchical path.

When you create your DXTVIEW, list the names of the fields you want to be visible. The fields you name must be described in the DXTPCB description and must be in a single hierarchical path. Your list must name at least one field, but need not name every field described in the path.

The names you use for these fields must either be those that are used in the DXTPCB description or they can be an alias for any field in the DXTVIEW. That alias must then be used when the field is referenced in an EXTRACT command. DataRefresher requires aliases if two or more fields with the same name are to be visible through your view. You must assign an alias to at least one of those fields to assign that field a unique name.

Specifying a path: Specify one path for your view description of a DXTPCB. Specify a path by indicating the name of its bottom segment with the SEGMENT keyword. When you specify the bottom segment, all segments in the path from the top level segment (root) to the bottom segment you specify are automatically included in the DXTVIEW.

After you specify the bottom segment, you can also specify the minimum segment. The minimum segment indicates the lowest segment in the file that must be present for the data to qualify for extraction. If you do not specify a minimum segment, MINSEG default to the segment you specified with the SEGMENT keyword. The complete path down to the specified SEGMENT is required for data extraction to occur.

For example, assume you are creating a view for a DXTPCB that was structured like this:

```
           COURSE
            |
          /   \|
         STUDENT  TEACHER
                   |
                  GRADE
```

There are two available paths in this structure. One starts at COURSE and goes to STUDENT and to GRADE. Another starts at COURSE and goes to TEACHER.
For the examples that follow, assume the segment occurrences of the first path, from COURSE to GRADE, look like this:

![Diagram of segment occurrences]

To create a DXTVIEW of the path from COURSE to STUDENT to GRADE:

1. SEGMENT=GRADE and MINSEG=MGRADE

   The entire path from COURSE to GRADE is included in the view. For data to qualify for extraction, all segments, COURSE, STUDENT, and GRADE, must be present. The qualified output rows for the segment occurrences shown in the figure above are:

   ENGLISH CURTIS A
   ENGLISH GAZA B

   The ENGLISH CHIN output row and the MATH output row do not qualify for extraction because they do not have a GRADE occurrence.

   **Note:** Without the MINSEG keyword, it defaults to GRADE.

2. SEGMENT=GRADE and MINSEG=STUDENT

   The entire path from COURSE to GRADE is included in the view. For data to qualify for extraction, COURSE and STUDENT segments have to be present. The qualified output rows for the segment occurrences shown in the figure above are:

   ENGLISH CURTIS A
   ENGLISH GAZA B
   ENGLISH CHIN

   The MATH output row still does not qualify for extraction because it does not contain a STUDENT segment.

3. SEGMENT=STUDENT

   In this case, the path only goes to STUDENT, so no GRADE data is extracted. Because no MINSEG is included, it defaults to STUDENT, and the STUDENT segment must be present for data to qualify for extraction. The qualified output rows for the segment occurrences shown in the figure above are:

   ENGLISH CURTIS
   ENGLISH GAZA
   ENGLISH CHIN
DXTVIEWs of an IMS PSB
The following views are of DXTPCBs provided with the sample data.

**IMSSTAFF VIEW**

```sql
CREATE DXTVIEW NAME=IMSSTAFF,
   DXTPSB=SAMPPSB1,
   DXTPCB=SAMPPCB1,
   SEGMENT=HIST,
   MINSEG=ORG,
   FIELD=*,
   DESC='HSAM, VIEW TO STAFF';
```

1. IMSSTAFF is the name of the view.
2. This view is over the DXTPSB named SAMPPSB1.
3. SAMPPCB1 is the name of the DXTPCB in SAMPPSB1 that this view looks at.
4. HIST is the segment specified, indicating that a path from the root segment (ORG) to the bottom segment (HIST) is included in this view.
5. ORG is the minimum segment specified, indicating that for a row of data to qualify for extraction, it need only contain the ORG segment.
6. FIELD=* indicates that all fields are chosen.

**IMSPROJECT view**

```sql
CREATE DXTVIEW NAME=IMSPROJECT,
   DXTPSB=SAMPPSB1,
   DXTPCB=SAMPPCB1,
   SEGMENT=PROJ,
   MINSEG=ORG,
   FIELD=*,
   DESC='HSAM, VIEW TO PROJ';
```

1. IMSPROJECT is the name of this view.
2. The segment specified is PROJ, indicating that a path from the root segment (ORG) to the bottom segment (PROJ) is included in this view.
3. ORG is the minimum segment specified, indicating that for a row of data to qualify for extraction, it need only contain the ORG segment.
4. FIELD=* indicates that all fields are chosen.

**IMSHIST view**

```sql
CREATE DXTVIEW NAME=IMSHIST,
   DXTPSB=SAMPPSB2,
   DXTPCB=SAMPPCB2,
   SEGMENT=HIST,
   MINSEG=ORG,
   FIELD=*,
   DESC='HSAM, VIEW TO HIST';
```

1. IMSHIST is the name of this view.
2. This view is over the DXTPSB named SAMPPSB2.
SAMPPCB2 is the PCB in SAMPPSB2 that this view looks at.

HIST is the segment specified, indicating that a path from the root segment (ORG) to the bottom segment (HIST) is included in this view.

ORG is the minimum segment specified, indicating that for a row of data to qualify for extraction, it need only contain the ORG segment.

FIELD=* indicates that all fields are chosen.

**Example with an alias:** This example illustrates how to use an alias in a DXTVIEW of a DXTPSB. A subset of the DXTPSB looks like this (keywords are eliminated to show you the structure only):

```
CREATE DXTPSB NAME=STORES,
   DXTPCB NAME=GROCERY,
   SEGMENT NAME=LOCATION,PARENT=0,
   FIELD NAME=NAME,,
   FIELD NAME=CITY,,
   FIELD NAME=PHONE,,
   SEGMENT NAME=DEPT,PARENT=LOCATION,
   FIELD NAME=NAME,,
   FIELD NAME= AISLE,,
   FIELD NAME= MANAGER,,
   . . . . . .
```

The following view of STORES includes the path down to DEPT and selects all fields:

```
CREATE DXTVIEW NAME=GROCVIEW,
   DXTPSB=STORES,
   DXTPCB=GROCERY,
   SEGMENT=DEPT, 1
   MINSEG=LOCATION,
   FIELDS=(STORENAME=LOCATION.NAME,CITY,PHONE,
        DEPTNAME=DEPT.NAME,AISLE,MANAGER); 2
```

1 The path, specified with SEGMENT=DEPT, goes from the root (LOCATION) to DEPT.

2 All the fields are specified, but not with FIELDS=* . Because two fields have the same name, you must assign an alias to at least one of them. In this case, aliases are assigned to both. The field NAME in the LOCATION segment is assigned the name STORENAME. The field NAME in the DEPT segment is assigned the name DEPTNAME. These aliases must be used in the EXTRACT command that uses this view description.
Chapter 4. Creating data descriptions and extract requests

This chapter describes the facilities provided by DataRefresher for creating data descriptions and extract requests, and tells you where you can obtain more information about each facility.

Structures Access Program

The Structures Access Program (SAP) allows the DataRefresher administrator to generate data description statements and extract requests based on stored information about IMS database definitions (DBDs), VSAM files, and physical sequential files.

For IMS database definitions, the SAP generates:

- CREATE DXTPSB statements
- CREATE DXTVIEW statements
- SUBMIT/EXTRACT statements

Chapter 10, “Extracting data using the SAP” on page 89 describes how to create data descriptions and extract requests. It also describes the SAP conventions and restrictions.

The following models are provided with DataRefresher for use by the SAP:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVRXKJDA</td>
<td>SAP skeleton containing the CREATE DXTPSB statement</td>
</tr>
<tr>
<td>DVRXKJFA</td>
<td>SAP skeleton containing the CREATE DXTFILE statement</td>
</tr>
</tbody>
</table>

Dictionary Access Program

If your site uses the IBM OS/VS DB/DC Data Dictionary, you can create DXTPSB descriptions using the DataRefresher Dictionary Access Program (DAP).

DAP uses the Program Access Facility of the IMS Data Dictionary to create data descriptions. To use DAP, you should have a working knowledge of the Data Dictionary.

Chapter 11, “Creating data descriptions using the DAP” on page 107 contains a description of how to generate data descriptions using DAP.

DataRefresher dialogs

There are two types of DataRefresher dialogs:

- Administrative Dialogs
- End User Dialogs
Both types of dialogs can be used in an MVS environment (MVS/ESA*) or a VM environment (VM/SP or VM/XA*). The Dialogs provide you with a series of panels for perform the following DataRefresher tasks:

- Creating data descriptions
- Storing data descriptions in the FDTLIB
- Creating extract requests
- Storing extract requests in the EXTLIB
- Creating the JCS to load the extracted data into the desired data target
- Submitting commands to DataRefresher

**Note:** Before you can use the Dialogs, you must carry out the actions described in the *DataRefresher Administration Guide* for administering the Dialogs.

### Administrative Dialogs

The DataRefresher Administrative Dialogs are designed for the experienced data processing user or database administrator, who has experience with:

- JCL or CP/CMS commands
- DataRefresher commands and statements
- The source database and target system
- The ISPF editor

The Administrative Dialogs provides a set of options and panels to help you:

- Create data descriptions and/or extract requests for the following sources:
  - DB2 databases
  - IMS DL/1 databases
  - SQL/DS databases
  - VSAM files
  - Physical sequential data sets
  - Self-defining files; Integration Exchange Format (IXF)
  - Non-IBM databases, via the generic data interface (GDI)
- Manage the JCL and JCS files that you use to:
  - Start UIM to validate data descriptions and extract requests and store them in the appropriate library
  - Start the REM to process relational extract requests
  - Direct the extracted data to its target
- Start the DAP
- Start the SAP
- Manage your dialog processing options
- Set up and administer the End User Dialogs

### End User Dialogs

The End User Dialogs make the extraction of data easier by providing you with a set of dialog panels to collect the information that DataRefresher needs to build an extract request.

The End User Dialogs are designed so that beginners or infrequent users of DataRefresher will be able to use DataRefresher without having to learn all the information required to create an extract request.
To use the End User Dialogs panels, you need to be experienced with Interactive System Productivity Facility (ISPF). A user working with the DataRefresher End User Dialogs can only extract data from a DataRefresher data source if the DataRefresher administrator has created the data descriptions and JCL which are required to describe and process the data to be extracted.

### Online DataRefresher commands

Table 5 lists the commands that can be used to create and run extract requests.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCANCEL</td>
<td>Cancels an extract request after it has been sent to the extract manager for processing.</td>
</tr>
</tbody>
</table>
| DCREATE | Creates one of the following data descriptions:  
  - DXTFILE  
  - DXTVIEW  
  - DXTPSB  
  - DXTPCB  
  - Datatype |
| DDELETE | Deletes data descriptions from the File Description Table Library (FDTLIB). |
| DLIST | Displays the following information:  
  - Extract request ID  
  - Extract request originator node and user ID  
  - Name of the file, PSB, or PCB specifying the extract data  
  - Time and date when request was submitted  
  - Output limit  
  - Priority number  
  - State of the extract request  
This information is also written to the DXTPRINT data set. |
| DPRINT | Displays a data description on the terminal and writes it to the DXTPRINT data set. |
| DPUNCH | Displays a data description on the terminal and writes it to the DXTPRINT and DXTPUNCH data sets. |
| DRUN | Runs the DEM to process a non-relational extract request. |
| DRUNR | Runs the REM to process a relational extract request. |
| DSEND | Creates an extract request and sends it for processing. |
| DSTATUS | Checks the status of an extract request, and displays the current status. |

For information about the syntax and use of these commands, see the DataRefresher Command Reference.
System editor

You can use your system editor to create DataRefresher data descriptions and extract requests based on the models supplied with DataRefresher. These models are all stored in:

- Data set DVR110.DVRJEDIE in MVS
- Macro library DVRJEDIE MACLIB in VM

All of the supplied models contain comments which provide a series of instructions on how to change the model to suit your requirements.

The following sections show the steps you take to create and process extract requests using the system editor. If you are using the system editor to create and process your extract requests use the following sections as guidelines. For information about the DataRefresher commands used in the section see DataRefresher Command Reference.

Extracting data from an IMS database

Carry out the following steps to extract data from an IMS database, and send it to:

- A relational database (DB2 or SQL/DS)
- A physical sequential database or CMS file
- IXF dataset or file

1. Create a DXTPSB description of the IMS database using the DVREDREP model.

2. Create a DXTVIEW of PCB using the DVREDRVP model.

3. Use the DVREDDXT model to run UIM to store the DXTPSB and DXTVIEW data descriptions in the FDTLIB.

4. Create your extract request using the DVREDEXT model. To send your data to:

   - A DB2 table, add the DVREDDJC model to the extract request to generate the JCS required to load the extracted data into the DB2 table.
   - An SQL/DS table, add the DVREDSJC model to the extract request to generate the JCS required to load the extracted data into the SQL/DS table.

**Note:** Make the following entry for the DBS parameter of the SUBMIT command, which is contained in the model:

- DBS=SQLDS For an SQL/DS target
- DBS=IXF For an IXF target

5. Run UIM to store the extract request and its JCS in EXTLIB using the model DVRDDEXT, which was also used to store the DXTPSB and DXTVIEW in the FDTLIB.

6. Run DEM to process the extract which is stored in the EXTLIB.
Extracting data from a VSAM data set

Carry out the following steps to extract data from a VSAM data set, and send it to:

- A relational database (DB/2 or SQL/DS)
- A physical sequential database or CMS file
- IXF dataset or file

1. Create a DXTFILE description of the VSAM data set using the DVREDREP model.

2. Create a DXTVIEW of DXTFILE using the DVREDRVF model.

3. Use the DVREDDXT model to run UIM to store the DXTPSB and DXTVIEW data descriptions in the FDTLIB.

4. Create your extract request using the DVREDEXT model. To send your data to:
   - A DB2 table, add the DVREDDJC model to the extract request to generate the JCS required to load the extracted data into the DB2 table.
   - An SQL/DS table, add the DVREDSJC model to the extract request to generate the JCS required to load the extracted data into the SQL/DS table.

   **Note:** Make the following entry for the DBS parameter of the SUBMIT command which is contained in the model:
   
   ```
   DBS=SQLDS        For an SQL/DS target
   DBS=IXF          For an IXF target
   ```

5. Run UIM to store the extract request and its JCS in EXTLIB using the model DVREDEXT, which was used to create the extract request.

6. Run DEM to process the extract which is stored in the EXTLIB.

Extracting data from a DB2 database

Carry out the following steps to extract data from a DB2 database, and send it to:

- A relational database (DB/2 or SQL/DS)
- A physical sequential database or CMS file
- IXF dataset or file

1. Create your extract request using the DVREDRXM model.

   **Notes:**
   
   a. Make the following entry for the DBS parameter of the SUBMIT command which is contained in the model:
   
   ```
   DBS=SQLDS        For an SQL/DS target
   DBS=IXF          For an IXF target
   ```

   b. To join the data extracted from two DB2 databases on the same host system, specify the join in the WHERE clause of the SUBMIT command which is contained in the model.

2. Write the JCL required to start REM and link to DB2, using the DVREDREM model.

3. If your target is:
   - A DB2 table, use the DVREDDJC model to generate the JCS required to load the data into a DB2 table.
• An SQL/DS table, use the DVREDSJC model to generate the JCS required to load the data into an SQL/DS table.

• A PSDS, CMS file, or IXF data set, go to the next step.

4. Use the JCL created model DVREDREM to run the REM to process the extract. The REM extracts the data and places it in the target table.

**Extracting data from an SQL/DS database**

Carry out the following steps to extract data from an SQL/DS database, and send it to:

• A relational database (DB/2 or SQL/DS)
• A physical sequential database or CMS file
• IXF dataset or file

1. Create your extract request using the DVREDRVS model.

   **Note:** For an IXF target use the DBS=IXF parameter in the SUBMIT command which is contained in the model.

2. Write the JCL required to start REM and link to DB2, using the DVREDREV model.

3. If your target is:
   • A DB2 table, use the DVREDDJC model to generate the JCS required to load the data into a DB2 table.
   • An SQL/DS table, use the DVREDSJC model to generate the JCS required to load the data into an SQL/DS table.
   • A CMS file, or an IXF data set or file, go to the next step.
   • A PSDS, use the DVREDRUM model to reformat the data using the Data Reformat Utility (DRU).

4. Use the DVREDRXV model to run the REM to process the extract. The REM extracts the data and loads it in the target table.

**Extracting data from an IMS database and a DB2 database**

Carry out the following steps to join data that has been extracted from an IMS database and a DB2 database, and send it to:

• A relational database (DB/2 or SQL/DS)
• A physical sequential database or CMS file
• IXF dataset or file

1. Create a DXTPSB description of the IMS database, using the DVREDREP model.

2. Create a DXTVIEW of the PCB using the DVREDRVP model.

3. Create a DXTFILE for the DB2 database to be accessed by the GDI exit using the DVREDREF model.

4. Write your GDI exit to extract from DB2 based on the sample DVRXOGRX provided in DVRSAMPE.

5. Create a DXTVIEW of the file containing the exit routine.

6. Use the DVREDDXT model to run the UIM to store the DXTPSB, DXTFILE, and DXTVIEW data descriptions in the FDTLIB.
7. Create your extract request using the DVREDEXT model. To send your data to:

- A DB2 table, add the DVREDDJC model to the extract request to generate the JCS required to load the extracted data into the DB2 table.
- An SQL/DS table, add the DVREDSJC model to the extract request to generate the JCS required to load the extracted data into the SQL/DS table.

**Note:** Make the following entry for the DBS parameter of the SUBMIT command which is contained in the model:

- `DBS=SQLDS` For an SQL/DS target
- `DBS=IXF` For an IXF target

To join the data extracted from two databases specify the join in the WHERE clause of the SUBMIT command which is contained in the model.

8. Run the UIM to store the extract request and its JCS in EXTLIB using the model DVRDEDEXT, which was used to store the PSB file and view in the FDTLIB.

9. Run the DEM to process the extract, which is stored in the EXTLIB.

**Extracting data from a DB2 database and a VSAM data set**

Carry out the following steps to join data that has been extracted from a DB2 database and a VSAM data set, and send it to:

- A relational database (DB/2 or SQL/DS)
- A physical sequential database or CMS file
- IXF dataset or file

1. Create a DXTFILE description of the VSAM data set, using the DVREDREP model.

2. Create a DXTVIEW of the data using the DVREDRVF model.

3. Create a DXTFILE for the DB2 database, using the DVREDREF model.

4. Write a GD1 exit routine to access the DB2 database based on the sample DVRXOGRX provided in DVRSAMPE.

5. Create a DXTVIEW of the file containing the exit routine.

6. Use the DVRREDXT model to run UIM to store the DXTPSB, DXTFILE, and DXTVIEW data descriptions in the FDLIB.

7. Write your extract request using the DVREDEXT model. To send your data to:

- A DB2 table, add the DVREDDJC model to the extract request to generate the JCS required to load the extracted data into the DB2 table.
- An SQL/DS table, add the DVREDSJC model to the extract request to generate the JCS required to load the extracted data into the SQL/DS table.

**Note:** Make the following entry for the DBS parameter of the SUBMIT command which is contained in the model:

- `DBS=SQLDS` For an SQL/DS target
- `DBS=IXF` For an IXF target
To join the data extracted from two databases specify the join in the **WHERE** clause of the **SUBMIT** command which is contained in the model.

8. Run the UIM to store the extract request and its JCS in EXTLIB using the model DVRDDEXT, that was used to store the PSB file and view in the FDTLIB.

9. Run DEM to process the extract, which is stored in the EXTLIB.

**Extracting data from an IMS database and a VSAM data set**

Carry out the following steps to join data that has been extracted from an IMS database and a VSAM data set, and send it to:

- A relational database (DB/2 or SQL/DS)
- A physical sequential database or CMS file
- IXF dataset or file

1. Create a DXTFILE description of the IMS database, using the DVREDREP model.

2. Create a DXTVIEW of the PCB using the DVREDRVP model.

3. Create a DXTFILE description of the VSAM data set, using the DVREDREP model.

4. Create a DXTVIEW of the VSAM data using the DVREDRVF model.

5. Use the DVREDDXT model to run UIM to store the DXTPSB, DXTFILE, and DXTVIEW data descriptions in the FDTLIB.

6. Write your extract request using the DVREDEXT model. To send your data to:

   - A DB2 table, add the DVREDDJC model to the extract request to generate the JCS required to load the extracted data into the DB2 table.
   - An SQL/DS table, add the DVREDSJC model to the extract request to generate the JCS required to load the extracted data into the SQL/DS table.

*Note:* Make the following entry for the DB5 parameter of the **SUBMIT** command which is contained in the model:

\[
\begin{align*}
\text{DB5} = \text{SQLDS} & \quad \text{For an SQL/DS target} \\
\text{DB5} = \text{IXF} & \quad \text{For an IXF target}
\end{align*}
\]

To join the data extracted from the IMS database and the VSAM data set, specify the join in the **WHERE** clause of the **SUBMIT** command which is contained in the model.

7. Run the UIM to store the extract request and its JCS in EXTLIB using the model DVRDDEXT, which was used to store the PSB file and view in the FDTLIB.

8. Run DEM to process the extract, which is stored in the EXTLIB.
Part 2. Using the Administrative Dialogs
Chapter 5. What are Administrative Dialogs?

DataRefresher Administrative Dialogs are intended for the data processing user, usually the database administrator, who has knowledge of the JCL and the ISPF editor. Using the Administrative Dialogs, you can:

- Extract data (options 1 and 2)
- Manage your JCL and JCS files (options 3 and 4)
- Set up the End User Dialogs for your users (option 5)

Working with Administrative Dialogs

When you log on to the Administrative Dialogs, the DataRefresher copyright panel is displayed. Press ENTER to display the Administrative Dialogs main menu panel:

```
DATAREFRESHER ADMINISTRATIVE DIALOGS

SELECT OPTION ---->
  1  EXTRACT       Build and maintain extract requests.
  2  DESCRIPTION   Build and maintain data description requests.
  3  JCL/JCS       Build and maintain Job Control Language.
  4  PROFILE       Specify Dialogs processing options.
  5  ADMINISTER    End User Dialogs Administration.

Press:  HELP key for information   ENTER to continue   END key to return
```

Type a number on the DataRefresher Administrative Dialogs main menu command line to display any of the following task panels:

1 EXTRACT
   Lets you select a dialog for building, updating, erasing, sending, importing, or exporting an extract request. It also lets you build a job to cancel, list, or check the status of an extract request in the EXTLIB.

2 DESCRIPTION
   Lets you select a dialog for building, updating, erasing, sending, importing or exporting a data description. It also lets you build a job to print, punch, or delete a data description request to the FDTLIB, or produce a DAP or SAP request.

3 JCL/JCS
   Lets you select a dialog for building, updating, importing, exporting, or erasing JCL or JCS.

4 PROFILE
   Lets you initiate the dialog for reviewing and changing your DataRefresher dialogs profile.

5 ADMINISTER
   Lets you initiate the dialog for completing the administrative tasks for DataRefresher End User Dialogs, which include, building the end user table, building the nickname table, and requesting source description information.
The following function keys are available on this menu and on most of the Administrative Dialogs panels:

HELP (F1) Displays a help panel for the menu or panel displayed.
ENTER Displays the next panel or menu, or completes an action returning you to the previous panel or menu.
END (F3) Deletes any changes made to the current panel and returns to the previous panel.

How you can use the Administrative Dialogs

The Administrative Dialogs provide models to help you build the data descriptions and extract requests you need. These models provide inline comments that help you specify the correct commands and values. The Administrative Dialogs also provide models of the kinds of JCL and JCS files that you need to use with your data descriptions and extract requests. You can edit any of the models to conform to your environment.

You can save your extract requests, data descriptions, JCL or JCS and use them for future extracts. In addition, you can perform administrative tasks that let users with a minimum knowledge of DataRefresher use the End User Dialogs through the Administrative Dialogs.

Using the Administrative Dialogs requires a knowledge of:

- DataRefresher commands and statements
- JCL (in MVS) and/or CP/CMS commands (in VM)
- Details about your systems that are used by DataRefresher

For more details about coding DataRefresher commands, JCL, JCS, and CP/CMS commands, see the DataRefresher Command Reference. Users interested in extracting data, but not in learning about DataRefresher, should use the End User Dialogs.

The Administrative Dialogs provide online help information to help you use the panels. Online help explains the fields on each panel to help you build a profile, maintain security, and administer end users.
Navigating through Administrative Dialogs

The Administrative Dialogs have three types of panels:

- Panels
- Display panels
- Edit panels

Navigating in panels: Typically in the Administrative Dialogs panels, you navigate forward from one panel to the next by using the ENTER key. To return to the panel you just came from, use the END key.

You cannot continue forward until all required fields on the current panel are filled in. If you do not fill in all required fields, an error message is displayed.

Navigating in display or edit panels: You can use ISPF scroll commands for display and edit panels. The following figure shows an example of such a panel. When the panel first displays on your screen, you are looking at the top of the table.

<table>
<thead>
<tr>
<th>COMMAND ===&gt;</th>
<th>SCROLL ===</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;TITLE OF DISPLAY/EDIT PANEL&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;TEXT OF PANEL HERE&gt;</td>
<td></td>
</tr>
<tr>
<td>******************************************* BOTTOM OF DATA *******************************************</td>
<td></td>
</tr>
</tbody>
</table>

The BOTTOM OF DATA line in the previous figure means that you are at the bottom of the panel. If the BOTTOM OF DATA line is not on your screen, there is additional information.

To see the additional information, scroll down until you reach the BOTTOM OF DATA line. You may change the default scroll amount in the SCROLL field. The new value becomes the default.

See Interactive System Productivity Facility Dialog Management Services for more information about scrolling and setting function keys.

Note: If you are unsure of your function key settings, use the KEYS command to display the current settings.
Chapter 6. Starting the Administrative Dialogs

This chapter describes how to start a session with the Administrative Dialogs. The
dialogs are started on MVS by invoking DVRELLGN CLIST and on VM by invoking
the DVREXUSR EXEC.

A description of how to set up the DVRELLGN CLIST and the DVREXUSR EXEC
is provided in DataRefresher Administration Guide.

Starting an MVS dialog session

Use the following steps to start an MVS DataRefresher dialogs session:

1. Run the ISPF logon procedure. This procedure sets up the ISPF environment
   and invokes the ISPF main menu.

2. Select the MVS DataRefresher dialogs option from the ISPF menu.

3. A message may be displayed asking whether you want to use the
   DataRefresher sample data. This message is only displayed if the
   DataRefresher sample data is available in your environment.

   Type Y and press ENTER to use the sample data, or press ENTER if you do
   not want to use the sample data.

4. A message may be displayed asking whether you want this dialog session to
   be an object sharing session. This message is only displayed if object sharing
   is available in your environment.

   Type Y if you want to use object sharing, or N if you do not want to use object
   sharing and press ENTER.

When the CLIST is complete, the DataRefresher copyright panel is displayed.
Press ENTER, to display the Administrative Dialogs main menu.

Starting a VM dialog session

Use the following steps to start a VM DataRefresher dialogs session:

1. Run the ISPF invocation EXEC.

   This EXEC should set up the ISPF environment and invoke the ISPF main
   menu; it assumes that the DataRefresher production disk has already been
   accessed.

2. Select the Administrative Dialogs option from the ISPF menu.

3. A message may be displayed asking whether you want to use the
   DataRefresher sample data. This message is only displayed if the
   DataRefresher sample data is available in your environment.

   Type Y and press ENTER to use the data, or press ENTER if you do not want
   to use the sample data.

4. A message may be displayed asking whether you want this dialog session to
   be an object sharing session. This message is only displayed if object sharing
   is available in your environment.
Type Y for yes to use object sharing, or N if you do not want to use object sharing and press ENTER.

When the EXEC is complete, the DataRefresher copyright panel is displayed. Press ENTER, to display the Administrative Dialogs Main Menu.
Chapter 7. Maintaining your JCL and JCS files

This chapter describes how to create and administer the JCL and JCS models shipped with DataRefresher. These models serve as a basis for building the JCL jobstreams and VM command streams used to invoke REM, UIM, SAP, and DAP. The models are stored in the data set or MACLIB allocated to DD name DVRJEDIE.

You can edit the DataRefresher default models, or create new objects using the models. You may want to:

**Edit the shipped models:**

- To enroll other users more easily.
  
The DataRefresher default models are automatically copied into a new user's dialog library when you run DVREXLIB or DVRELNRL to enroll a user. If you edit the default models, new users will receive the modified models automatically.

  Renamed models are not automatically copied. The Import/Export dialog described in “Providing edited JCL files” on page 135 is used to copy them into a user's DataRefresher dialogs library.

- Because you do not need new objects.
  
  If you are sending DataRefresher requests for the same type of data to a single system, you may need only one copy of the appropriate JCL.

- If it is more convenient.
  
  You may decide that the administration involved in tracking new object models is too difficult.

**Create new objects because:**

- You need separate JCL or JCS files.
  
  To extract the same type of data from multiple systems, you might need a different JCL file for each system from which you want to extract data. For example, you may need different job cards or other specific information for each system.

- You may have different requirements in the future.
  
  If you keep the models, you always have a master copy should you require them.
Creating and editing JCL/JCS models

To create and edit a new object based on a model, and to edit a model, follow the steps in this section.

1. Log on to the Administrative Dialogs.

```
SELECT OPTION ===> 3
  1 EXTRACT Build and maintain extract requests.
  2 DESCRIPTION Build and maintain data description requests.
  3 JCL/JCS Build and maintain Job Control Language.
  4 PROFILE Specify Dialogs processing options.
  5 ADMINISTER End User Dialogs Administration.

Press: HELP key for information ENTER to continue END key to return
```

Figure 18. DataRefresher Administrative Dialogs main menu

2. Select option 3, JCL/JCS, from the 'DataRefresher Administrative Dialogs' main menu.

The 'Build and Maintain JCL Files' panel is displayed.

```
SELECT OPTION ===> 1
  1 EDIT, BUILD, or ERASE a JCL/JCS file.
  2 IMPORT or EXPORT a JCL file.

```

Figure 19. Build and Maintain JCL File panel

3. Select option 1,

```
EDIT, BUILD, or ERASE a JCL/JCS file
```

to display a list of the JCL/JCS models and JCL/JCS files.

4. The action you take depends on whether you are creating a new file or editing a model:

- To edit a model or file, type an S in the SEL column next to the appropriate model and press ENTER. Continue with the next step, step 5 on page 67.

- To create a new JCL or JCS file from a model, type an M in the SEL column next to the model which is to be used as a base for the new file, and press ENTER.

**Note:** You can delete a model or JCL/JCS file by typing an E in the SEL column next to the appropriate model and pressing ENTER.

When you create a new JCL/JCS file, the 'Identify New JCL/JCS Name' panel is displayed. This panel prompts you to enter a name and description for the file.

In the following example the JCL is given the name NEWJCL and the description New JCL using DVREDREM.
COMMAND => IDENTIFY NEW JCL/JCS NAME

Create and enter a NAME for the new JCL/JCS
The DESCRIPTION field is optional, but highly recommended.

NAME  DESCRIPTION
NEW     NEWJCL     New JCL using DVREDREM
MODEL  DVREDREM   V100 DEFAULT JCL MODEL -- REM EXECUTION (MVS)
      TYPE      JCL

Figure 20. Identify New JCL/JCS Name panel

5. An ISPF edit session is started, so that you can edit the JCL/JCS model or file.

For an explanation of what to specify in the file, see the comments in the JCL/JCS model. These comments are guidelines.

You may have site-specific information you need to add, or find that some of the suggested changes may not apply to your site.

6. Press END (F3), when you have completed your changes to the file, to store the file.

Note: If you are the DataRefresher administrator and want to use object sharing, you must have write access to the file or data set allocated to DVRSTABL. The LIBDEF for the DD name DVRSTABL is set in DVREXUSR or Dvrellg, as described in DataRefresher Administration Guide.

JCL and JCS models supplied with DataRefresher

Table 6 describes the JCL models supplied with DataRefresher, while Table 7 on page 68 describes the JCS models supplied with DataRefresher. The tables also provide some edit instructions for using the models.

<table>
<thead>
<tr>
<th>Model name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVREDDXT</td>
<td>This JCL starts the DXTINPUT procedure to invoke the UIM to store extract requests, data descriptions, and for the PRINT, PUNCH, STATUS, CANCEL and LIST commands. It also routes a source data description request (GETDEF) to UIM during an End User Dialogs Administration dialog.</td>
</tr>
<tr>
<td>DVREDDAP</td>
<td>This JCL routes requests for data descriptions to the Dictionary Access Program (DAP) and invokes DAP to process the request.</td>
</tr>
<tr>
<td>DVREDREM</td>
<td>This JCL invokes the REM to process an extract request on a DB2 data source.</td>
</tr>
<tr>
<td>DVREDREV</td>
<td>This JCL invokes the REM in VM to run an extract request on an SQL/DS database.</td>
</tr>
<tr>
<td>Model name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| **DVREDDJC** | This JCS is used to place data into an existing DB2 table in MVS. You need to specify a JCS file when you want DataRefresher to generate an output job. The JCS statements include a *C0 and *E0, which show DataRefresher where to insert the control deck and extracted data in the output job.  
**DVREDDJC:**  
- Invokes the DB2 load utility  
- Contains the extracted data that is placed into a DB2 table, through the DB2 load utility |
| **DVREDBBC** | This JCS is used for placing data into a created or replaced DB2 table in MVS. You need to specify a JCS file when you want DataRefresher to generate an output job. The JCS statements include a *TC, *C0, and *E0, which show DataRefresher where to insert the SQL CREATE or DELETE statement, the control deck, and extracted data in the output job.  
**DVREDBBC:**  
- Creates a DB2 table or erases the contents of an existing DB2 table  
- Invokes the DB2 load utility  
- Contains the extracted data that is placed into a DB2 table, through the DB2 load utility |
| **DVREDSJC** | This JCS is used for loading data into SQL/DS in VM. You need to specify a JCS file when you want DataRefresher to generate an output job. The JCS statements include a *C0 and *E0, which show DataRefresher where to insert the control deck and extracted data in the output job.  
**DVREDSJC:**  
- Invokes the SQL/DS load utility  
- Contains the extracted data that is loaded into a SQL/DS table, through the SQL/DS load utility |
| **DVREDEJM** | This JCS is used to run DataRefresher’s MIT utility in MVS. You need to specify a JCS data set when you want DataRefresher to generate an output job. The JCS statements include a *C0 and *E0, which show DataRefresher where to insert the control deck and extracted data in the output job.  
**DVREDEJM:**  
- Starts the MVS Master Index Table (MIT) utility to add to or update the MIT during End User Dialogs administration  
- Contains the data used to update the MIT |
Table 7 (Page 2 of 2). Supplied JCS models

<table>
<thead>
<tr>
<th>Model name</th>
<th>Description</th>
</tr>
</thead>
</table>
| DVREDEJV   | This JCS runs DataRefresher’s MIT utility in VM. You need to specify a JCS data set when you want DataRefresher to generate an output job. The JCS statements include a *CD and *EO, which show DataRefresher where to insert the control deck and extracted data in the output job. DVREDEJV:  
  - Starts the VM Master Index Table (MIT) utility to add or update the MIT during End User Dialogs administration  
  - Contains the data used to update the MIT  
This JCS is used to run the Data Reformat Utility on MVS when data has been transferred over a network. |
| DVREDRUM   | You need to specify a JCS file when you want DataRefresher to generate an output job. The JCS statements include a *CD and *EO, which show DataRefresher where to insert the control deck and extracted data in the output job. DVREDRUM:  
  - Starts the Data Reformat Utility (DRU) on an MVS system  
  - Contains the data that the DRU reformats |
| DVREDRUV   | This JCS runs the DRU in VM. You need to specify a JCS file when you want DataRefresher to generate an output job. The JCS statements include a *CD and *EO, which show DataRefresher where to insert the control deck and extracted data in the output job. This JCS is intended to be used with a DRU running in batch mode. To run the DRU interactively, use DVREXBV4 EXEC, described in the program directory for the VM base product. DVREDRUV:  
  - Starts the Data Reformat Utility (DRU) in VM  
  - Contains the data that the DRU reformats  
This JCS is used to run the Data Reformat Utility on VM when data has been transferred over a network. |

Directing output in MVS

For information about directing print output according to the conventions of your installation, see your system administrator.

When dialogs and UIM are run in the same MVS system you can direct your punch output to an existing member of a partitioned data set by providing the name of the partitioned data set in a DXTPUNCH statement. For example, if you want to place the output in member SALCOMM of partitioned data set DEPTTAB, include the following statement in your JCL:

```
//UIM.DXTPUNCH DD DSN=DEPTTAB(SALCOMM),DISP=SHR
```

To make your data description punch output available on dialogs after TSO notifies you that your job has finished, uses the Import function of DataRefresher Administrative Dialogs to move the member you need into your dialogs library.
When you want your extract output directed to a relational table on the same system from which it was extracted, you need to provide JCS to run the LOAD utility. TSO notifies you when your load job has completed if you included the NOTIFY parameter in the JOB statement in your JCS.

To see the extracted data after TSO notifies you that your load job is complete, you can create queries on the loaded table using Query Management Facility or IBM DATABASE 2 Interactive (DB2I).

When you want your extract output directed to a different system to the system from which it was extracted you can include the JES ROUTE-FOR-EXECUTE in your JCS. The extracted output will need to be reformatted at its destination using the Data Reformat Utility.

### Directing output in VM

To print your output include a route statement. For example:

```plaintext
ROUTE PRINT vmnode/vmuser
```

Where `vmnode/vmuser` refers to the system node you are on and your user ID. To route your output to the system printer using the following statement:

```plaintext
//UIM.DXTPRINT DD SYSOUT=A
```

VM returns the print output to your VM reader. To see the print output, browse the file in your reader list.

To copy punch output into your dialogs library provide a route punch statement. For example:

```plaintext
ROUTE PUNCH vmnode/vmuser
```

Where `vmnode/vmuser` refers to the system node and your user ID. To route your output to the system punch use the following statement:

```plaintext
//UIM.DXTPUNCH DD SYSOUT=B
```

You can use the CMS READ command to read the output from your reader into a file. Once you have loaded the output into a file, with a filetype of COPY or MACRO, use the CMS MACLIB command to add the file to your import/export library (MACLIB), so you can import it. The format of this command is:

```plaintext
MACLIB subcom libname fn
```

Where:

- `subcom`  
  - ADD if the member does not exist in the import/export library, and REP if it does

- `libname`  
  - Is the name of the IMPORT/EXPORT library that contains the member file containing the punch output

- `fn`  
  - Is the name of the file that contains punch output

**Note:** Use the Import function of dialogs to load the object into your dialogs library. MACLIB commands do not register the object in your dialogs library.
To direct your extract output to a relational table on the same system from which it was extracted, provide JCS to run the LOAD utility.

To see your extracted data after the load job is complete, you can create queries on the loaded tables using QMF.
Chapter 8. Extracting data from a non-relational source

This chapter describes how to extract data from an IMS database, a VSAM data set, or physical sequential data set and how to load the extracted data into another object using the DataRefresher Administrative Dialogs.

Table 8 lists all of the tasks you perform when using the Dialogs to extract and output the data.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Create the JCL required to store the data descriptions in the FDTLIB (File Description Table Library), based on model JCL supplied with DataRefresher.</td>
</tr>
<tr>
<td>2</td>
<td>Create the DXTPSB or DXTFILE description, based on a model supplied with DataRefresher.</td>
</tr>
<tr>
<td>3</td>
<td>Create a DXTVIEW, based on a model supplied with DataRefresher.</td>
</tr>
<tr>
<td>4</td>
<td>Send the data descriptions to the FDTLIB.</td>
</tr>
<tr>
<td>5</td>
<td>Create JCS based on a model supplied with DataRefresher.</td>
</tr>
<tr>
<td>6</td>
<td>Create an extract request based on a model supplied with DataRefresher.</td>
</tr>
<tr>
<td>7</td>
<td>Send extract request and JCS to the ETLIB (Extract Request Library).</td>
</tr>
<tr>
<td>8</td>
<td>Create a job to run DEM to extract the data and send it to the target object.</td>
</tr>
<tr>
<td>9</td>
<td>Run the DEM to process the extract.</td>
</tr>
</tbody>
</table>

The following sections show how to perform the tasks listed above, and tell you where to find more detailed information about performing the tasks.

Step 1. Creating JCL using a model

In this step you create JCL to invoke the UIM to store your data descriptions in the FDTLIB (File Description Table Library).

1. Select option 3 JCL/JCS, from the 'Administrative Dialogs' main menu.

The 'Build and Maintain JCL Files' menu shown in Figure 21 is displayed.

```
SELECT OPTION --->

1  EDIT, BUILD, or ERASE a JCL/JCS file.
2  IMPORT or EXPORT a JCL/JCS file.
```

Figure 21. Build and Maintain JCL Files menu
2. Select option 1 Edit, Build, or Erase a JCL/JCS file.

A list of all JCL/JCS models and any JCL/JCS files you have already created is displayed. A example of the panel displayed is shown in Figure 22.

```
COMMAND ===>  LIST OF JCL FILE NAMES          ROW 1 OF 11
            SCROLL ===>  PAGE
Press:  HELP key for information  ENTER to continue  END key to return

SEL : s = select to EDIT,  e = ERASE,
       m = use as a MODEL to create a new JCL/JCS

SEL  JCL/JCS NAME  TYPE  DESCRIPTION
  -  DVREDDAP  JCS  V1RO DEFAULT JCL MODEL -- DAP EXECUTION
  -  DVREDDJC  JCS  V1RO DEFAULT JCL MODEL -- DB2 LOAD EXECUTION
  -  DVREDDXT  JCL  V1RO DEFAULT JCL MODEL -- UIM EXECUTION
  -  DVREDJIM  JCS  V1RO DEFAULT JCL MODEL -- MIT LOAD EXECUTN (MVS)
  -  DVREDJRV  JCS  V1RO DEFAULT MODEL EXEC -- MIT LOAD EXECUTN (VM)
  -  DVREDMEM  JCL  V1RO DEFAULT JCL MODEL -- REM EXECUTION (MVS)
  -  DVREDREV  JCL  V1RO DEFAULT MODEL EXEC -- REM EXECUTION (VM)
  -  DVREDRUM  JCS  V1RO DEFAULT JCL MODEL -- DRU EXECUTION (MVS)
  -  DVREDRUV  JCS  V1RO DEFAULT MODEL EXEC -- DRU EXECUTION (VM)
  -  DVREDJSC  JCS  V1RO DEFAULT JCL MODEL -- SQL/DS LOAD EXECUTION

******************************************************************************
Press:  HELP key for information  ENTER to continue  END key to return
```

Figure 22. List of JCL File Names panel

3. To create a new JCL file from the JCL model for UIM execution, DVREDDXT:
   - Type M in the SEL column next to the name of the model.
   - Press ENTER

   **Note:** You could select and edit one of the models or files by using S to select the model or file. However, any changes which you make remain in the model or file. Copying the model enables you to create a separate file, so that you can copy the model several times.

The Identify New JCL/JCS Name panel, shown in Figure 23, is displayed.

```
COMMAND ===>  IDENTIFY NEW JCL/JCS NAME
Create and enter a NAME for the new JCL/JCS
The Description field is optional, but highly recommended.

          NAME  DESCRIPTION
NEW  -

MODEL  DVREDDXT  V1RO DEFAULT JCL MODEL -- UIM EXECUTION
        TYPE  JCL

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

Figure 23. Identify New JCL/JCS Name panel

4. Type the name to be used to identify the JCL in the Name field, and a description of the function of the JCL in the Description field.

5. Press ENTER.
An ISPF edit session is started, so that you can edit the JCL file. For an explanation of what to change to suit your requirements see the comments in the JCL file.

When you have finished editing the file press the END key to save the file and return to the 'List of JCL File Names' panel shown in Figure 22 on page 74.

---

**Step 2. Creating a DXTPSB or DXTFILE description using a model**

In this step you create a DXTPSB description of the IMS database, or a DXTFILE description of a VSAM or SDS data set.

To create the DXTPSB or DXTFILE description:

1. Return to the DataRefresher 'Administrative Dialogs' main menu.
   
   To access this menu from the 'List of JCL File Names' panel, press the END key twice.
   
2. Select option 2, Description.
   
3. Select option 1 from the 'Build and Maintain Data Description Requests' panel, to create a new data description request.
   
   The panel shown in Figure 24 is displayed.

---

**LIST OF DATA DESCRIPTION REQUESTS**

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>SCROLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>PRESS: HELP key for information ENTER to continue END key to return</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SEL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>s = select and EDIT, e = ERASE, m = use as a MODEL to create a new request, x = SEND the data description request</td>
<td></td>
</tr>
</tbody>
</table>

---

**Figure 24. List of Data Description Requests panel**

4. The model you select depends on whether you are extracting data from an IMS or non-IMS source:

**IMS database**
   
   Select the model DVREDREP to create a DXTPSB

**VSAM or SDS data set**

   Select the model DVREDREF to create a DXTFILE

To select the model:

a. Type M in the SEL column, next to the model name.

b. Press ENTER.
The 'Identify New Request Name' panel is displayed. The panel shown in Figure 25 on page 76 is for the DXTPSB model. The panel displayed for the DXTFILE model is similar to this panel.

![Identify New Request Name panel](image)

**Figure 25. Identify New Request Name panel**

5. Type the name to be used to identify the DXTPSB or DXTFILE in the Name field, and a description of the function of the DXTPSB or DXTFILE in the Description field.

6. Press ENTER.

An ISPF edit session is started, so that you can edit the DXTPSB or DXTFILE. The DXTPSB model contains the UIM command CREATE DXTPSB, while the DXTFILE model contains the UIM command CREATE DXTFILE. For information about these commands see the *DataRefresher Command Reference*.

When you have completed the DataRefresher description, press the END key to save the file and exit the editing session.

The 'JCL Review - Data Description Request' panel, shown in Figure 26, is displayed. You use this panel to specify the JCL file that invokes UIM to place the data description in the FDTLIB.

![JCL Review - Data Description Request panel](image)

**Figure 26. JCL Review - Data Description Request**
7. Type the name of the JCL file you created in Step 1, Creating JCL using a model.

8. Press ENTER.

You return to the 'List of Data Description Requests' panel, shown in Figure 24 on page 75, and the data description you just created is displayed in the list of data descriptions.

Step 3. Creating a DXTVIEW using a model

In this step, you create a DXTVIEW to describe to DataRefresher the particular data that you are extracting from the IMS database.

To create the DXTVIEW:

1. Type M in the SEL column, next to the model name DVREDRVP on the 'List of Data Description Requests' panel, shown in Figure 24 on page 75.

2. Press ENTER.

The 'Identify New Request Name' panel, shown in Figure 25 on page 76, is displayed.

3. Type a name for the DXTVIEW in the Name field, and a description of the function of the DXTVIEW in the Description field.

4. Press ENTER.

An ISPF edit session is started, so that you can edit the View. The DXTPSB or DXTFILE contains the UIM command CREATF DXTVIEW. For information about this command see the DataRefresher Command Reference.

When you have completed the DXTVIEW, press the END key to save the file and exit the editing session.

5. Press ENTER to return to the 'List of Data Description Requests' panel.

Step 4. Sending data descriptions to the FDTLIB

When the data descriptions and JCL have been created, you can send the DXTPSB or DXTFILE description and the DXTVIEW to the FDTLIB.

To send the data descriptions and DXTVIEW to the FDTLIB:

1. Type X in the SEL column of the 'List of Data Description Requests' panel, next to the name of the DXTPSB, DXTFILE, or DXTVIEW description you created in Step 2 or 3. This panel is shown in Figure 25 on page 76.

2. Press ENTER.

The 'JCL Review - Data Description Request' panel is displayed so that you can confirm the name of the JCL file.

3. Press ENTER, to confirm the name of the JCL file, or type the name of the file and press ENTER.

An ISPF edit session is started on the JCL file that contains the JCL and the data description. You can check that the file is correct, and make changes to the file in this session.

4. Press the END key when you are ready to exit the editing session.
The 'Send Confirmation' panel is displayed to confirm that you want to send the job to the FDTLIB.

5. Type Y and press ENTER to send the job.

   **Note:** After pressing ENTER, you may have to press the PA2 or CLEAR key to complete the confirmation.

Check your diagnostic output messages to see whether the job completed successfully. The messages should contain the following lines:

```
DVRU0661 CREATE COMMAND PROCESSING COMPLETED SUCCESSFULLY
   (LINE 8, COLUMN 1). 002 D
DVRU0101 UIM PROCESSING IS COMPLETE. RETURN CODE 0. 002 S
```

If your job has processed successfully, the DXTPSB or DXTFILE now resides in the FDTLIB. Repeat the above steps, to send the DXTVIEW to the FDTLIB.

---

**Step 5. Creating JCS using a model**

In this step, you create JCS that can be used to create a table and load the extracted output into the table.

To create the JCS:

1. Return to the DataRefresher 'Administrative Dialogs' main menu.

   To return to the menu from the 'List of JCL File Names' panel, press the END key **twice**.

2. Select option 3 JCL/JCS, from the 'Administrative Dialogs' main menu.

   The 'Build and Maintain JCL Files' menu shown in Figure 21 on page 73 is displayed.

3. Select option 1 on the Build and 'Maintain JCL Files' panel to build the JCS.

   The 'List of JCL Files Names' panel shown in Figure 22 on page 74 is displayed.

4. Select the appropriate model from the panel by entering M in the SEL column next to the model name.

   For example, the model DVREDDBC creates a DB2 table and loads the data into the table.

   The 'Identify New JCL/JCS Name' panel, shown in Figure 23 on page 74, is displayed.

5. Type the name of the JCS file in the Name field, and a description of the function of the JCS in the Description field.

6. Press ENTER.

   An ISPF edit session is started, so that you can edit the JCS file. For an explanation of what to change to suit your requirements, see the comments in the JCS file.

   When you have finished editing the file, press the END key to save the file and return to the 'List of JCL File Names' panel.
Step 6. Creating an extract request using a model

This task involves creating an extract request to identify the data that DataRefresher is to extract.

To create the extract request:

1. Return to the DataRefresher 'Administrative Dialogs' main menu.
   To return to the menu from the 'List of JCL File Names' panel, press the END key twice.
2. Select option 1 Extract, from the 'Administrative Dialogs' main menu.
   The 'Build and Maintain Extract Requests' is displayed.
3. Select option 1 from the 'Build and Maintain Extract Requests' menu to create a new extract request.
   The 'List of Extract Requests' panel, shown in Figure 27, is displayed.

<table>
<thead>
<tr>
<th>LIST OF EXTRACT REQUESTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND ==&gt;</td>
</tr>
<tr>
<td>Press: HELP key for information ENTER to continue END key to return</td>
</tr>
<tr>
<td>SEL  s = select and EDIT, e = ERASE, m = use as a MODEL to create a new request x = SEND the extract request</td>
</tr>
<tr>
<td>REQUEST NAME</td>
</tr>
<tr>
<td>DWAREX</td>
</tr>
<tr>
<td>DWAREXXM</td>
</tr>
<tr>
<td>************************************************************************* BOTTOM OF DATA *************************************************************************</td>
</tr>
</tbody>
</table>

Figure 27. List of Extract Requests

4. Select the model you require:
   - Type M in the SEL column next to name of the extract request.
   - Press ENTER.
   The 'Identify New Request Name' panel shown in Figure 25 on page 76 is displayed.

5. Type the name of the Extract Request in the Name field, and a description of its function in the Description field.

6. Press ENTER.
   An ISPF edit session is started so that you can edit the JCS file. For an explanation of what to change to suit your requirements, see the comments in the model.
   When you have finished editing the file press the END key to save the file.
   The 'JCL Review - Extract Request' panel, shown in Figure 28 on page 80, is displayed.
7. Type the following information on this panel:
   - The name of the JCL to invoke the UIM, created in Step 1
   - The name of the JCS to create the table, and load the data into the target database, created in Step 5
   - The JCS DD name

   **Note:** If you want to display a list of existing JCL files, enter ? in the JCL name field.

8. Press ENTER to save the changes and return to the 'List of Extract Requests' panel shown in Figure 27 on page 79.

---

**Step 7. Sending an extract request and JCS to the EXTLIB**

The three components needed to send an extract request to the EXTLIB have now been created.

To send the extract request:

1. Select the extract request to be submitted by entering X in the SEL column next to the request name, on the 'List of Extract Requests' panel shown in Figure 27 on page 79.

   The 'JCL Review - Extract Request' panel, shown in Figure 28, is displayed.

2. Confirm that the following information shown on the panel is correct:
   - Execution JCL name
   - Load JCS name
   - JCS DD name

   The JCS DD name must match the name specified as the value of the JCS keyword in the extract request.

3. Press ENTER.

   An ISPF edit session is started so you can review the job stream, which is a concatenation of the JCL, JCS, and the extract request.
Press the END key when you are ready to exit the editing session. The 'Send Confirmation' panel is displayed so that you confirm that the job is to be sent to the EXTLIB.

4. Enter Y in the input field to send the job to the EXTLIB.

Note: After pressing ENTER, you may have to press the PA2 or CLEAR key to complete the confirmation.

The job is now sent to the EXTLIB await processing by the DEM. When the job has completed, check the output messages; they should contain the messages similar to the ones shown below:

```
12:00:00 06/01/91 DVRU1201 THE EXTRACT REQUEST WITH EXTID=STAFFEXT, USERID=USERID, AND NODE=NODEID IS ADDED TO THE EXTLIB SUCCESSFULLY.
12:00:00 06/01/91 DVRU0101 UIM PROCESSING IS COMPLETE. RETURN CODE 0.
```

---

**Step 8. Creating a job to run the DEM**

To create a job to run the DEM:

1. Allocate a new sequential data set, outside the Administrative Dialogs, with fixed blocked (FB) record format and a logical record length (LRECL) of 80 bytes. Set the number of directory blocks to 0 so a sequential file is allocated.

You can also use an existing data set or a member of an existing partitioned dataset with the same logical record length.

2. Start an ISPF edit session on the data set.

3. In this new data set, create the job stream shown in Figure 29 on page 82. This job stream runs DEM in an IMS batch environment using the IMS sample data.

Notes:

a. The example shown below is for an IMS data source.

b. Alter the values in **bold** type to reflect your specific user information.
Figure 29. Batch Job to run DEM

JCL Notes:

1. Your job card must conform to your installation standards.
2. The PSB keyword of the DLIBATCH PROC names the actual IMS PSB.
3. This is the IMS execution library.
4. These two data sets name the IMS PSB and DBD libraries.
5. These two data sets name the DataRefresher-supplied IMS sample data.
6. Includes the FDTLIB and EXTLIB to be used in this extract.
7. Includes the name of the DXTPSB description of the IMS PSB.
Step 9. Running the DEM to process the extract

To process the extract, enter the command SUBMIT on the command line. For details about the syntax of this command see the DataRefresher Command Reference.

When the job has completed, the output messages should contain messages similar to the following:

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Message Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVRS0405</td>
<td>THE EXTRACT REQUEST WITH EXTID=STAFFEXT, USERID=USERID, AND NODE=NODEID HAS BEGUN EXECUTION. 012 S</td>
</tr>
<tr>
<td>DVRS0504</td>
<td>THE EXTRACT REQUEST WITH EXTID=STAFFEXT, USERID=USERID, AND NODE=NODEID HAS RUN TO COMPLETION, WITH 9 ROWS OF EXTRACT OUTPUT. 013 E</td>
</tr>
<tr>
<td>DVRS0522</td>
<td>THE EXTRACT REQUEST WITH EXTID=STAFFEXT, USERID=USERID, AND NODE=NODEID HAS BEEN DELETED FROM THE EXTLIB. 014 S</td>
</tr>
<tr>
<td>DVRS0104</td>
<td>DEM PROCESSING IS COMPLETE. RETURN CODE 0. 015 S</td>
</tr>
</tbody>
</table>

When the DEM has successfully completed processing your extract request, the JCS output job will be submitted for execution.
Chapter 9. Extracting data from a relational source

This chapter describes how to extract data from a relational database, such as DB2 or SQL/DS, and how to load the extracted data into one of DataRefresher’s targets using the DataRefresher Administrative Dialogs.

Table 9 describes all of the tasks you perform when using the Dialogs to extract and load the data.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Create JCS based on a model supplied with DataRefresher.</td>
</tr>
<tr>
<td>2</td>
<td>Create an extract request based on a model supplied with DataRefresher.</td>
</tr>
<tr>
<td>3</td>
<td>Create a job to run REM to extract the data and send it to the target object.</td>
</tr>
<tr>
<td>4</td>
<td>Run the REM to process the extract.</td>
</tr>
</tbody>
</table>

The following sections in the chapter describe how to perform the tasks shown in the table, and tell you where to find more detailed information about performing the tasks.

Step 1. Creating JCS using a model

In this step, you create JCS which can be used to create a table and load the extracted output into the table.

To create the JCS:

1. Select option 3 JCL/JCS, from the 'Administrative Dialogs' main menu.
   The 'Build and Maintain JCL Files' menu shown in Figure 21 on page 73 is displayed.

2. Select option 1 Edit, Build, or Erase a JCL/JCS File, on the 'Build and Maintain JCL Files' panel to build the JCS.
   The 'List of JCL Files Names' panel shown in Figure 22 on page 74 is displayed.

3. Select the appropriate model or file from the panel by typing M in the SEL column next to the model name and pressing ENTER.
   For example, the model DVREDDBC creates a DB2 table and loads the data into the table for relational to relational extracts.
   The 'Identify New JCL/JCS Name' panel, shown in Figure 23 on page 74, is displayed.

4. Type the name to be used to identify the JCS in the Name field, and a description of the function of the JCS in the Description field.

5. Press ENTER.
An ISPF editing session is started so that you can edit the JCS file. For an explanation of what to change to suit your requirements, see the comments in the JCS file.

When you have finished editing the file press END to save the file and return to the 'List of JCL File Names' panel shown in Figure 22 on page 74.

---

**Step 2. Creating an extract request using a model**

This task involves creating an extract request to determine which data DataRefresher is to extract.

To create the extract request:

1. Select option 1 Extract, from the 'Administrative Dialogs' main menu to display the 'Build and Maintain Extract Requests' menu.
2. Select option 1 Create, from the 'Build and Maintain Extract Requests' panel to create a new extract request.

The 'List of Extract Requests' panel, an example of which is shown in Figure 30, is displayed.

<table>
<thead>
<tr>
<th>SEL REQUEST NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVMEXT</td>
<td>GENERAL</td>
<td>VIR0 DEFAULT MODEL -- UIM/DEM EXTRACT</td>
</tr>
<tr>
<td>DVMSUM</td>
<td>RELATNM</td>
<td>VIR0 DEFAULT MODEL -- REM EXTRACT (NVS)</td>
</tr>
<tr>
<td>DVMKEX</td>
<td>RELATNM</td>
<td>VIR0 DEFAULT MODEL -- REM EXTRACT (VM)</td>
</tr>
</tbody>
</table>

---

**Figure 30. List of Extract Requests**

3. Select the model you want to copy and edit by typing M in the SEL column next to the model name, and pressing ENTER.

The 'Identify New Request Name' panel shown in Figure 25 on page 76 is displayed.

**Note:** You could select and edit one of the models or files by using S to select the model or file. However, any changes which you make remain in the model or file. Copying the model enables you to create a separate file, so that you can copy the model several times.

4. Type the name to be used to identify the Request in the field Name, and a description of the function of the Request in the Description field.

5. Press ENTER.

An ISPF editing session is started so that you can edit the JCS file. For an explanation of what to change to suit your requirements see the comments in the model.

When you have finished editing the file press END to save the file.
The 'JCL Review - Extract Request' panel, shown in Figure 31 on page 87, is displayed.

```
JCL REVIEW - EXTRACT REQUEST

COMMAND ===>

This is the JCL name associated with the request.
You may change the following names for this request
WITHOUT affecting the defaults established in your profile.

  Execution JCL   ====> UIMRUN
  Load JCS       ====> DB2RUN
  JCS DDname     ====> DB2LOAD

To display a list of existing JCL files,
enter a ? in the JCL file name field.

If you are SENDING a request, after pressing the ENTER key
you may have to press the PAZ or CLEAR key to complete the dialog.

Press: HELP key for information  ENTER to continue  END key to return
```

Figure 31. JCL Review - Extract Request

6. Type the following information on this panel:
   - The name of your execution JCL
   - The name of your load JCS
   - The JCS DD name

   **Note:** If you want to view a list of existing JCL files, enter ? in one of the JCL
   name fields.

7. Press ENTER.

   You return to the 'List of Extract Requests' panel, shown in Figure 30 on
   page 86.

---

### Step 3. Creating a job to run the REM

To create a job to run the REM:

1. Allocate a new sequential data set, outside the Administrative Dialogs, with:
   - 0 directory blocks
   - Fixed blocked (FB) record format
   - Logical record length (LRECL) of 80 bytes

   You can also use an existing dataset or a member of a partitioned dataset with
   the same logical record length.

2. Start an ISPF edit session on the data set.

3. In this new data set, create the job stream to run REM.

   The DataRefresher model DVREDREM contains the JCL required to run REM
   when extracting data from a DB2 source. If you are extracting data from an
   SQL/DS source, you should base the job stream on the model contained in the
   DVREDREV model.
Step 4. Running the REM to process the extract

If you are using the REM on:

- MVS, run the job created in Step 3 using the TSO SUBMIT command.
- VM, send the job created in Step 3 to the CMS batch machine using the CP Punch command.

When the job has completed, check that the output messages are similar to the following:

```
DVRS0405  THE EXTRACT REQUEST WITH EXTID=STAFFEXT, USERID=USERID,
          AND NODE=NODEID HAS BEGUN EXECUTION.  012 S

DVRS0504  THE EXTRACT REQUEST WITH EXTID=STAFFEXT, USERID=USERID,
          AND NODE=NODEID HAS RUN TO COMPLETION, WITH 9 ROWS OF
          EXTRACT OUTPUT.  013 E

DVRS0522  THE EXTRACT REQUEST WITH EXTID=STAFFEXT, USERID=USERID,
          AND NODE=NODEID HAS BEEN DELETED FROM THE EXTLIB.
          014 S

DVRS0104  REM PROCESSING IS COMPLETE. RETURN CODE 0.  015 S
```

When REM has successfully completed processing your extract request, the JCS output job will be submitted for execution.
Chapter 10. Extracting data using the SAP

The Structures Access Program (SAP) allows the DataRefresher administrator to generate data description statements and extract requests based on stored information for IMS database definitions (DBDs), VSAM data sets, and physical sequential data sets.

This chapter describes the tasks that the DataRefresher administrator must perform to create the data description statements and extract requests. It also describes the SAP conventions and restrictions.

Conventions and restrictions

This section contains information about the rules and limitations of using the SAP with COBOL and PL/I structures.

Conventions for SAP data structure conversion

The following table describes the conventions the SAP observes in converting data structures to DataRefresher CREATE and SUBMIT/EXTRACT statements:

| If a picture field contains | • Edit characters (for example, Z), the field is treated as a character field
|                           | • Only the P, S, V, and 9 symbols, the field is treated as a numeric field |
| If any attributes are      | • Not provided for a field; the SAP uses the standard default rules for PL/I and COBOL
|                           | • Not supported (such as LIKE or any dynamic sizes); they may cause error conditions |
| In PL/I                     | • The data structure must be defined between columns 2 and 72 of the PL/I declare library |
| In COBOL                   | • The data definitions must fall within standard COBOL input areas
|                           | • A hyphen (−) in COBOL data names is changed to an underscore (_). |
Restrictions to SAP processing
To generate data description statements and an extract request for your DL/I
database, VSAM file, or sequential file, the SAP has to process the data definitions
using PL/I or COBOL structures. If you are generating statements for a DL/I
database, the SAP also needs information from an IMS DBD.

The following information outlines the restrictions as they apply to these libraries
and DBDs:

General restrictions
• You must be able to compile the PL/I and COBOL structures as individual
  objects.
• The IMS DBD generation program must be able to process the database
definitions.
• Any data type not included as either PL/I or COBOL is ignored, but its length is
  accounted for.
• If either character or picture data exceeds the maximum lengths, the SAP
  ignores that field; however, the SAP calculates the length and alignment to
  position later items correctly.
• If any field names are duplicated, the first occurrence of the name is used as
  provided. Each occurrence of the name after the first is altered in the output to
  make the name of the field a unique name.
• If the COBOL or PL/I structure defines one or more fields that overlay the key
  field defined by the IMS DBD, the SAP generates both the DBD key field and
  the structure’s field names.

IMS
• The SAP supports logical DBDs. The structure name must match the name on
  the DBD. For a logical DBD, this structure must represent the logical view of
  the data returned by DL/I.
• Secondary indexes are not supported.
• The segment names in the DBD must match the member names in a COBOL
  or PL/I structure partitioned data set.

PL/I structures
• The PL/I compiler must be able to process the PL/I structures. The PL/I
  structures must begin with a DECLARE statement and end with a semicolon (;).
• DCL (DECLARE) members can include other members using the %INCLUDE
  statement only if they exist in the same data set.
• The SAP analyzes only the first DECLARE (at the beginning of the data
  structure). The SAP ignores all succeeding DCLs.
• SAP uses the default PL/I rules for calculating field offsets. Take care when
  trying to align datatypes to byte and word boundaries.
• SKIP, PAGE, PRINT, and NOPRINT statements are permitted, but they are ignored.
• The LIKE attribute is not supported.
• Precompiler functions are not supported. If they are used, they either cause an
  incorrect analysis or a serious problem is reported.
• MARGINS are assumed to be 2 and 72.

• Standard defaults are assumed. The DEFAULT statement is not permitted.

• Only the following data types are explicitly supported:

<table>
<thead>
<tr>
<th>Data type</th>
<th>Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHARACTER(n)</td>
<td>n is in the range 0 to 254</td>
</tr>
<tr>
<td>CHARACTER(n) varying</td>
<td>n is in the range 0 to 32767</td>
</tr>
<tr>
<td>GRAPHIC(n)</td>
<td>n is in the range 0 to 127</td>
</tr>
<tr>
<td>GRAPHIC(n) varying</td>
<td>n is in the range 0 to 16383</td>
</tr>
<tr>
<td>BINARY FLOAT(n)</td>
<td>n is in the range 22 to 53</td>
</tr>
<tr>
<td>DECIMAL FLOAT(n)*</td>
<td>n is in the range 0 to 16</td>
</tr>
<tr>
<td>BINARY FIXED(a,b)</td>
<td>a is in the range 16 to 31, and b is 0</td>
</tr>
<tr>
<td>DECIMAL FIXED(a,b)*</td>
<td>a is in the range 1 to 15, and b 0 to 15</td>
</tr>
<tr>
<td>PICTURE</td>
<td>The total length must be less than 254</td>
</tr>
<tr>
<td>REAL numeric items</td>
<td>None</td>
</tr>
</tbody>
</table>

* SAP will accept a DECIMAL FLOAT value without n, by defaulting to a half-word (four byte) field. SAP also accepts a DECIMAL FIXED value without a,b, by defaulting to a DECIMAL FIXED(5, 0).

COBOL structures
• The COBOL compiler must be able to compile the COBOL structures as an object. The COBOL structures must begin with a 01 level and represent one database segment.

• Elements that follow a COBOL OCCURS DEPENDING ON statement, but are not subordinate to the OCCURS clause, cannot have a SYNCHRONIZE(ALIGN) clause.

• The USAGE INDEX and the USAGE POINTER are not supported. INDEX (POINTER) items are omitted from the output.

• P picture items are ignored.

• COMP-3 items are limited to 18 digits. COMP items are limited to 9 digits. Items longer than these limits are omitted from the output.

• Leading signs are not supported for numeric items. Numeric items are changed to CHAR if a leading sign is found.

• A nonzero scale (the use of the V picture) is omitted for COMP items.

• The RENAMES (level 66) option is not supported. Processing ends if one is found.

• *CBL, *CONTROL, SKIP, and EJECT are permitted but ignored.

• "D" debug lines are always treated as comments.

• Only the following data items are supported:

<table>
<thead>
<tr>
<th>Data type</th>
<th>Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY</td>
<td>Where the length is in the range 0 to 254</td>
</tr>
<tr>
<td>DISPLAY-1</td>
<td>Where the length is in the range 0 to 127</td>
</tr>
<tr>
<td>COMP</td>
<td>Where the length is less than 9 (digit positions)</td>
</tr>
<tr>
<td>COMP-1</td>
<td>none</td>
</tr>
<tr>
<td>COMP-2</td>
<td>none</td>
</tr>
<tr>
<td>COMP-3</td>
<td>Where the length is less than 18 (digit positions)</td>
</tr>
</tbody>
</table>
Specifying SAP execution information

Before you can specify SAP execution information, you must determine which data definition describes the source data for the DataRefresher extract and whether the data definition is in the IMS DBD libraries or a data set containing COBOL or PL/I language structures.

Starting the SAP

Either of the following DataRefresher JCL skeletons can be used to start SAP:

**DVRXKJFA** Invokes the SAP Command Generator to generate DataRefresher CREATE DXTFILE statements from COBOL or PL/I language structures

**DVRXKJDA** Invokes the SAP Command Generator to generate DataRefresher CREATE DXTPSB statements from IMS DBD elements and the underlying COBOL or PL/I structures

SAP generates accompanying CREATE DXTVIEW and SUBMIT/EXTRACT commands using the appropriate JCL skeleton. Both skeletons are stored in the DataRefresher Dialogs skeleton library and should be tailored for your specific environment during installation. For information about tailoring these JCL skeletons, see the DataRefresher Program Directory.

You must specify the information about the IMS DBD elements and data structures that you are using in the skeletons. To incorporate the specifications, you can use:

- SAP Dialogs panels
- An editor to modify a copy of the skeleton JCL

These methods are described in the following sections.

Using the SAP Dialog panels

You can start the SAP Dialogs from:

- The DataRefresher Administrative Dialogs main menu
- The ISPF main menu (if an option has been set up by the system administrator)

This section takes you through SAP Dialogs panels using the DataRefresher Administrative Dialogs. The sample data shown in this section, is provided with DataRefresher.

1. Select option 2 DESCRIPTION from the 'DataRefresher Administrative Dialogs' main menu.

The 'Build and Maintain Data Description Requests' panel shown in Figure 32 on page 93 is displayed.
BUILD AND MAINTAIN DATA DESCRIPTION REQUESTS

SELECT OPTION ===> 3

1  EDIT, CREATE, ERASE or SEND a data description request.
2  IMPORT or EXPORT a data description request.
3  PRINT or PUNCH data description(s).
4  DELETE data description(s) from the FDTLIB.
5  Invoke the SAP (Structure Access Program).

If you have the appropriate Program(s) installed:
6  Invoke the DAP (Dictionary Access Program).

Figure 32. Build and Maintain Data Description Requests panel

2. Select option 5, from the 'Build and Maintain Data Description Requests' panel, to start SAP Dialogs.

The 'Create DataRefresher Design' panel, shown in Figure 33, is displayed.

CREATE DAREFRESHER DESIGN

COMMAND ===> 

Select the type of data description to be built. ===> 1

1  FILE
2  PSB

Figure 33. Create DataRefresher Design panel

3. From the 'Create DataRefresher Design' panel, select one of the following options:

1  FILE Generates the table design for a VSAM or sequential file (a CREATE DXTFILE statement). When you select this option the 'Create DXTFILE for VSAM or SEQ File' panel, shown in Figure 34 on page 94, is displayed.

2  PSB Generate the table design for an IMS DL/I database (a CREATE DXTPSB statement). When you select this option the 'Create DXTPSB for DL/I Database' panel, shown in Figure 35 on page 94, is displayed.

4. The action taken in this step depends on the option selected in Step 3. If you selected:

- Option 1, to create a DXTFILE, use the panel displayed to type the information for the DXTFILE. When you have completed the fields on the panel, press ENTER. Figure 34 on page 94 shows the panel with example of the entries you would make.
CREATE DXTFILE FOR VSAM OR SEQ FILE

Command >>>

DECLARE COPY LIBRARY:
PROJECT >>>>
GROUP >>>>
TYPE >>>>
MEMBER >>>>
OTHER PDS >>>> 'DVR110.DVRSAMPE(DVRECSDP)'

DXTFILE TYPE >>>> SEQ (ESDS, KSDS, or SEQ)
LANGUAGE >>>> PLI (PLI or COBOL)

Do you want to validate the
selected Data Sets and Members?
Enter Yes or No >>>> YES

Press: HELP key for information ENTER to continue END key to return

Figure 34. Create DXTFILE for VSAM or Sequential File panel

- Option 2, to create a DXTPSB, use the panel displayed to type the
  information for the DXTPSB. When you have completed the fields on
  the panel, press ENTER. Figure 35 shows the panel with examples of
  the type of entries you would make.

CREATE DXTPSB FOR DL/I DATA BASE

Command >>>

IMS DBD SOURCE:
PROJECT >>>>
GROUP >>>>
TYPE >>>>
MEMBER >>>>
OTHER PDS >>>> 'DVR110.DVRSAMPE(DVRECSDD)'

DECLARE COPY LIBRARY:
PROJECT >>>>
GROUP >>>>
TYPE >>>>
OTHER PDS >>>> 'DVR110.DVRSAMPE(DVRECSDD)'

LANGUAGE >>>> COBOL (PLI or COBOL)

Do you want to validate the
selected Data Sets and Members?
Enter Yes or No >>>> YES

Press: HELP key for information ENTER to continue END key to return

Figure 35. Create DXTPSB for DL/I Database panel

When you press ENTER, the 'Create Commands Target Library' panel, shown
in Figure 36 on page 95, is displayed.

You use this panel to identify the library in which you want to generate and
store the CREATE DXTFILE or DXTPSB command and the SUBMIT/EXTRACT
command, and whether you want to replace the existing PDS members.
5. Type the following information in the 'Create Commands Target Library' panel, and press ENTER:

- **DataRefresher Create Command Target Library information.**

  To use the sample data supplied with DataRefresher type the following, depending on the option selected in Step 3 on page 93:

<table>
<thead>
<tr>
<th>Option 1</th>
<th>Option 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Other PDS</strong></td>
<td><strong>'DVR110.DVRIMEXE.DVRECSDD'</strong></td>
</tr>
</tbody>
</table>

- **Optional - Target Library Member... field**

  To use the sample data supplied with DataRefresher, type the following entry depending on the option selected in Step 3 on page 93:

<table>
<thead>
<tr>
<th>Option 1 FILE</th>
<th>Option 2 PSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVRECSBP</td>
<td>DVRECSUB</td>
</tr>
</tbody>
</table>

**Note:** If you make an entry in this field the 'Submit/Extract and Table Load Details' panel, shown in Figure 37 on page 96, is displayed after you press ENTER.

If you do not make an entry in this field, the 'Submit or Save the Generated JCL' panel is displayed, and you should continue from Step 7 on page 96

- **Specify whether existing members are to be replaced.**

6. If you made an entry for the optional target library member in Step 5, the 'Submit/Extract and Table Load Details' panel, shown in Figure 37 on page 96, is displayed.
### SUBMIT/EXTRACT and TABLE LOAD Details

<table>
<thead>
<tr>
<th>Command ===</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TARGET SYSTEM TYPE</td>
<td>(DB2, SQLDS, IXF or FILE)</td>
</tr>
<tr>
<td>TARGET NODE ID</td>
<td></td>
</tr>
<tr>
<td>EXTRACT DATA NAME</td>
<td>(Blank for DB2 or SQL/DS)</td>
</tr>
<tr>
<td>OUTPUT FORMAT</td>
<td>(ERCOIC or SOURCE)</td>
</tr>
<tr>
<td>REPLACE DB2/SQLDS TABLE</td>
<td>(YES or NO)</td>
</tr>
<tr>
<td>DB2 or SQL/DS USER ID</td>
<td></td>
</tr>
<tr>
<td>DB2 or SQL/DS TABLE NAME</td>
<td></td>
</tr>
<tr>
<td>DB2 or SQL/DS JCS NAME</td>
<td></td>
</tr>
</tbody>
</table>

Press: HELP key for information  ENTER to continue  END key to return

---

**Figure 37. Submit/Extract and Table Load Details panel**

Use the panel to specify the target table and its location and characteristics. If you are using the DataRefresher example you would make the following entries in this panel:

<table>
<thead>
<tr>
<th>Field</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target System Type</td>
<td>DB2</td>
</tr>
<tr>
<td>Target Node ID</td>
<td>nodename</td>
</tr>
<tr>
<td>Output Format</td>
<td>Source</td>
</tr>
<tr>
<td>Replace DB2/SQLDS Table</td>
<td>Yes</td>
</tr>
<tr>
<td>DB2 or SQL/DS User ID</td>
<td>userid</td>
</tr>
<tr>
<td>DB2 or SQL/DS Table Name</td>
<td>DB2TABLE</td>
</tr>
<tr>
<td>DB2 or SQL/DS JCS Name</td>
<td>DB2DDNAME</td>
</tr>
</tbody>
</table>

Press ENTER when you have types your entries in the panel. The 'Submit or Save the Generated JCL' panel, shown in Figure 38, is displayed.

---

### SUBMIT or SAVE the Generated JCL

<table>
<thead>
<tr>
<th>Command ===</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you want to CREATE the DataRefresher commands now or save the JCL for later execution.</td>
<td>(YES or NO)</td>
</tr>
<tr>
<td>1 Execute the JCL now</td>
<td></td>
</tr>
<tr>
<td>2 Save the JCL for later execution</td>
<td></td>
</tr>
</tbody>
</table>

SAVE JCL ISPF LIBRARY:

- PROJECT ===
- GROUP ===
- TYPE ===
- MEMBER ===
- OTHER PDS ===

REPLACE MEMBERS === YES  (REPLACE EXISTING MEMBER - YES OR NO)

Press: HELP key for information  ENTER to continue  END key to return

---

**Figure 38. Submit or Save Generated JCL panel**

7. The entry you make for this panel depends on whether the job is to be run immediately:
Option 1
Submits the generated JCL to SAP Command Generator.

Option 2
Saves the generated JCL. Specify the library and member where you want to store the JCL and whether you want to replace an existing member.

If you are using the DataRefresher example select option 2 and type the following entries in this panel:

MEMBER YES
OTHER PDS 'DVR110.DVREMEXE.DVRECSJL'

---

**Editing the skeleton JCL**

The two JCL skeletons (DVRXKJFA for DXTFILE statements and DVRXKJDA for DXTPSB statements) are stored in the DataRefresher Dialogs skeleton library.

**Note:** The skeletons cannot be accessed from the DataRefresher Dialogs. You must access them from outside DataRefresher to copy them and edit the copies.

You can use SAP Dialogs, shown in "Using the SAP Dialog panels" on page 92 to generate a copy of the JCL containing the information you need to invoke SAP. If you save this JCL, shown in Step 7 on page 96, you can edit and resubmit the JCL to the SAP to generate data descriptions and extract requests based on other structures.

Figure 39 shows the JCL skeleton for creating a CREATE DXTFILE statement for a VSAM data set or a physical sequential data set. Figure 41 on page 100 shows the JCL skeleton for creating a CREATE DXTFILE statement for a VSAM data set or a physical sequential data set.

```plaintext
//@USER JOB(,,BINRI),&ZUSER,USER=&ZUSER,,CLASS=A,
// MSGLEVEL=(1,1),MSGCLASS=X
/*NOTIFY ***NODEID***.&ZUSER
/*ROUTE PRINT ***NODEID***.&ZUSER
<<<<<<<<<DVRXKJFA<<<<<<<<<<<
<<<<<<<<<DVRM8LIBE MESSAGE LIBRARY CONTAINS THE SAP MESSAGES USED
/* in the Dialogs and SAP messages used for the sysprint
/* output.
/*
/* THOSE STATEMENTS WHICH MUST BE MODIFIED ARE
/* FLAGGED BY ENCLOSING THE PART TO BE MODIFIED
/* IN TRIPLE ASTERISK. FOR EXAMPLE:
/*
/* DSN=***DXTPRE***
/*
```

*Figure 39 (Part 1 of 2). JCL for generating a CREATE DXTFILE statement*
Figure 39 (Part 2 of 2). JCL for generating a CREATE DXTFILE statement

1. The DVRFL procedure sets up the environment required by the SAP. The PROC is executed by the IFG0 EXEC DVRFL command.

2. The INTSO inline data set invokes the SAP Command Generator.

3. The INVARS inline data set defines the parameters that the SAP uses to generate data description statements and an extract request.

4. The &SRCLIB variable identifies the structure library. Assign this variable a value either by using the dialogs or by editing the file. This variable should have the same value as DES2DSN.

98 MVS and VM User's Guide
The definitions of the parameters in the JCL model required to generate a CREATE DXFILE statement are:

- SRC_TYPE: SRC_TYPE=1 for structure (non-DBD) data
- DES2DSN: Segment structure data set name (copy lib)
- DES2MEM: Structure member name
- DES2LAN: Source language (PLI or COBOL)
- DES2FTY: Source file type (ESDS, KSDS, or SEQ)
- DES2REP: Replace the target data set? (Yes or No)
- LDJCルドN: Output target data set name
- LDJCルドEM: Output target member for DXTPSB/DXTVIEW commands
- LDJCルドEMX: Output target member for SUBMIT/EXTRACT data
- LDJCルドSTP: Output target type (DB2, SQLDS, IXF, or FILE)
- LDNODEID: SUBMIT node ID
- LDJCルドN: SUBMIT DD name
- LDFORMAT: SUBMIT output format (EBCDIC or SOURCE)
- LDREPM: Replace target relational table? (Yes or No)
- LUSERID: Create relational table user ID
- LDOUTTBIN: Create relational table name

If you followed the example in "Using the SAP Dialog panels" on page 92 to generate JCL for generating DataRefresher file data descriptions, the values of SAP input in the saved JCL shown in Figure 40.

```
//STEP.INVARS DD *
SRC_TYPE = 'I';
DES2DSN = '"DVR110.DVRSAMPE"';
DES2MEM = '"DVRECSDP"';
DES2LAN = 'PLI';
DES2FTY = 'SEQ';
DES2REP = 'YES';
LDJCルドN = '"DVR110.DVRIMEXE"';
LDJCルドEM = '"DVRECSDP"';
LDJCルドEMX = '"DVRECSBP"';
LDJCルドSTP = '"DB2"';
LDJCルドN = '"DB2DDONAM"';
LDNODEID = 'nodeid';
LDFORMAT = 'SOURCE';
LDREPM = 'YES';
LUSERID = 'userid';
LDOUTTBIN = 'DB2TABLE';
```

Figure 40. Example of SAP input to CREATE DXFILE
Figure 41 (Part 1 of 2). JCL for generating a CREATE DXTPSB statement
Figure 41 (Part 2 of 2). JCL for generating a CREATE DXTPSB statement

1. The DVRDBD procedure sets up the environment required by the SAP. The PROC is executed by the IPFGG0 EXEC DVRDBD command.

2. The INTSO inline data set invokes the SAP Command Generator.

3. The INVARS inline data set defines the parameters that SAP uses to generate data description statements and an extract request.

4. The &DBDSRC variable identifies the DBD library. Assign this variable a value either by using the dialogs or by editing the file.

   This variable should have the same value as DES1DN1.

5. The &SRCLIB variable identifies the structure library. Assign this variable a value either by using the dialogs or by editing the file. This variable should have the same value as DES1DN2.

The definitions for the parameters in the JCL model for the CREATE DXTPSB are:

- **SRCTYPE** = SRCTYPE=2 for DBD data
- **DES1MEM** = IMS DBD source member name
- **DES1DN1** = IMS DBD source data set name
- **DES1LAN** = Source language (PLI or COBOL)
- **DES1DN2** = IMS segment structure data set name (Copy Lib)
- **LDREPMEM** = Replace output target data set? (Yes or No)
- **LDCLDN1** = Output target data set name
- **LDCLMEM** = Output target member for DXTPSB/DXTVIEW commands
- **LDCLMEX** = Output target member for SUBMIT/EXTRACT data
- **LDCLSTP** = Output target type (DB2, SQLDS, IXF, or FILE)
- **LDCLDDN** = SUBMIT DD name
- **LDNODEID** = SUBMIT node ID
- **LDFORMAT** = SUBMIT output format (EBCDIC or SOURCE)
- **LDTREP** = Replace relational table? (Y or N)
- **LDUSERID** = Create relational table user ID
- **LDOUTBTN** = Create relational table name

If you have followed the example in “Using the SAP Dialog panels” on page 92 and saved the generated JCL, SAP input values in the saved JCL are shown in Figure 42 on page 102.
Submitting the JCL for batch processing

When you have generated or edited the JCL for invoking the SAP, submit the completed JCL in one of the following ways:

- From the SAP dialogs, select option 1, Execute the JCL now, on the 'Submit or Save the Generated JCL panel' (see Step 7 on page 96).
- While viewing the JCL with an editor, enter the SUBMIT command.

Using SAP output

The SAP generates three types of DataRefresher data description statements:

- CREATE DXTFILE (for VSAM or sequential files)
- CREATE DXTPSB (for IMS DL/I databases)
- CREATE DXTVIEW (for DXTVIEWs)

The SAP can optionally generate an extract request for the data source described by the input structure.

Note: Always review SAP output before you use it. You can use the error or warning messages that appear in the SYSPRINT output (the Command Generator report) to correct any deviations in the output data descriptions.

For any type of output member (data description or extract request statement), you can:

1. Edit the partitioned data set member containing the commands
2. Invoke the UIM to store the commands in the proper library (FDTLIB or EXTLIB)
3. Import the commands into the DataRefresher Dialogs library
Editing SAP output
You can edit the data description statements or the extract request with any editor that can edit partitioned data set members.

Before you submit the data descriptions or extract request to the UIM, check the output from the SAP for:

- Unwanted information.
  The SAP creates the UIM commands for your designated files and PSBs using all the DBD and language structures information available. As a result, the output may describe more segments, fields, or PCBs than you need for your extract requests.

- Missing information.
  Information may be missing from the DBD or language structure because it is not normally carried in these structures. An example is the DDNAME for a file. SAP specifies in the SYSPRINT output (SAP Command Generator report) that information is missing from the structure.

- Erroneous information.
  Information from the SAP could be erroneous because the structure was created or maintained incorrectly, or includes nonstandard representation of data.

- Information in the wrong format.
  When the SAP generates DataRefresher data descriptions for segments that include arrays, the SAP generates a separate path for each element in the array. In this type of data description, the parent segment is repeated for each element in the array, with only the array element changing for each path. If you want the data description to contain one path containing all occurrences in the array, you must edit the generated data description statement.

Invoking UIM to store SAP output
To store the data descriptions in the FDTLIB, specify the partitioned data set members containing the data descriptions as the DXTIN data set in the JCL used to invoke the UIM.

To store the extract request in the EXTLIB, specify the partitioned data set member containing the extract request as the DXTIN data set in the JCL used to invoke the UIM.

Importing SAP output into the DataRefresher dialog library
You can import the data descriptions and extract request into the DataRefresher Administrative Dialogs environment as model data descriptions or shared objects.

For more information about importing objects, see “Enrolling MVS DataRefresher dialogs users” on page 149.
Error handling

Error messages from the SAP Dialogs are displayed on the screen. If you want more information about the error message, press the HELP (F1) key.

Error messages from the SAP Command Generator are written to SYSPRINT. The SAP handles errors based on the return codes (rc) as follows:

Severe errors (rc ≥ 12)
Abnormal termination with problem diagnosis information written to SYSPRINT.

Figure 43 shows an example ISPF transaction log message for an abnormal termination:

```
*** ISPF TRANSACTION LOG ***
USERID: USERNAME DATE: 91/12/25
START OF ISPF LOG - - - - SESSION # 1 -------------------------------------------------
 TSO - COMMAND - - ALLOC F(ISPF) DA('USERNAME.SAP.OUTPUT')
 ***** DIALOG ERROR ***** - DVRXJ908 - SAP SEVERE ERROR
 - A SEVERE ERROR HAS OCCURRED DURING THE EXECUTION OF A SAP DIALOG.
 - INFORMATION DETAILING THE CAUSE OF THE ERROR IS LISTED BELOW.
 - THIS SAP DIALOG WILL BE TERMINATED WHEN YOU PRESS THE ENTER KEY.
 -
 - PLEASE TAKE THIS INFORMATION TO YOUR DXT SUPPORT PERSONNEL.
 -
 - MODULE :
 - SERVICE : FTINCL
 - PARM : DVRXJFK
 - RETURN CODE : 20
 -
 - ISPF183
 - IMBED FILE 'LOSTSKEL' DOES NOT EXIST, DVRXJFK RECORD-47
END OF ISPF LOG - - - - SESSION # 1 -------------------------------------------------
```

Figure 43. Example of ISPF transaction Log

Other errors (4 < rc < 12)
The SAP Command Generator writes message information to SYSPRINT, and the SAP Command Generator terminates processing.

Warning errors (0 < rc ≤ 4)
The SAP Command Generator writes message information to SYSPRINT and continues processing.

Figure 44 on page 105 shows an example of an exception report that would be written to SYSPRINT.
Figure 44. Example of Exception Report

For more information about SAP messages, refer to DataRefresher Messages and Codes.
Chapter 11. Creating data descriptions using the DAP

If your site uses the IBM OS/VS DB/DC Data Dictionary, you can create DXTFIL and PSB descriptions using the DataRefresher Dictionary Access Program (DAP).

The DAP uses the Program Access Facility of the IMS Data Dictionary to create data descriptions. To use the DAP, you should have a working knowledge of the Data Dictionary.

When creating data descriptions with the DAP you must follow the steps shown in Table 10.

<table>
<thead>
<tr>
<th>Table 10. DAP Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step</strong></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

Step 1. Creating the input for the DAP

You communicate in one of the following ways:

- Using the Data Dictionary EXECUTE command.
- Using a Batch Job
- Using the DataRefresher Administrative Dialogs

Using the EXECUTE command

The DAP passes this command to the Data Dictionary to request data definition information. For a description of the EXECUTE command, see DataRefresher Command Reference.

Using the DAP in batch

You can provide input to the DAP as part of the DDINPUT data set of your JCL. Figure 45 on page 108 shows an example of the JCL to provide the input to the DAP.
Figure 45. JCL for running the DAP in batch

If you need to route your jobs across systems, you must include either JES2 or JES3 statements in this JCL. The job routing statements in the example are JES2 statements. For more information about JES2 and JES3 statements and their syntax, consult your JCL documentation.

**jobname JOB**

Starts the job to invoke the DAP. When writing this statement, replace **jobname** by your jobname, and add the rest of the parameters for the statement.

**ROUTE XEQ**

Only use this statement if the DAP is run on a system other than the one from which you are submitting this job.

When writing the statement, replace **nodeid** with the node ID of the system running the DAP.

**ROUTE PRINT**

Sends the print output back to your system. Only use this statement if DataRefresher is run on a system other than the one on which you are submitting this job.

When writing the statement, replace **nodeid** with the node ID of the system running the DAP, and replace **userid** with your user ID on the printer system.

**ROUTE PUNCH**

Sends the punch output back to your system. Only use this statement if DataRefresher is run on a system other than the one from which you are submitting the UIM job.

When writing the statement, replace **nodeid** with the node ID of the system running the DAP.
STEP1 EXEC
Invokes the appropriate IMS module to run the Dictionary as an IMS application program. The parameters passed to the IMS module are:

DLI Specifications a batch run
DBDIMSBD Identifies the Dictionary module
DBDIMSOS Identifies the Dictionary module's PSB

For a description of the remaining parameters see the section on the DLIBATCH cataloged procedure in IMS/VS Version 2 System Programming Reference.

STEPLIB
The concatenation of the:

- IMS resident library (IMSVS.RESLIB)
- Data Dictionary resident library (DBD.IMSVS.PGMLIB),
- Library containing the DAP (DVR110.DVRLOAD).

The Dictionary module is connected to the DAP via the Dictionary's Program Access Facility.

DF5RESLB
The library containing the SVC modules that are required by IMS. For more information, see the IMS/VS Version 2 System Programming Reference.

IMS
The concatenation of the following libraries:

- The PSB library, DBD.IMSVS.PSBLIB
- The DBD library, DBD.IMSVS.DBDLIB

The PSB library specifies the PSB for the job, DBDIMSOS, and the DBD library contains the DBDs for the Dictionary databases.

IEFRDER
The IMS log file. You must include a statement for the log, even if you are not going to use it.

When you do not use the log, the DUMMY form of the IEFHDER statement can be used. For more information about the DLIBATCH procedure see the IMS/VS Version 2 System Programming Reference.

DDPSBLIB and DDBDDBLIB
DDPSBLIB is a PSB library. DDBDDBLIB is a DBD library. While the DAP uses neither of these data sets, DDPSBLIB and DDBDDBLIB statements must be present in the JCL and must represent real PSB and DBD libraries.

DTEDD, SEGDD, DBSDD, PCBDD, SYRDD, and EXTDD
These are the Data Dictionary databases. Each holds all the information on one or more categories of Dictionary subjects.

DXTDUMP
Receives the diagnostic information DAP produces in response to the DEBUG keyword. Always provide a DXTDUMP statement, whatever your specified debugging level.
When writing this statement, you must include the parameters for defining the data set as an output device (generally, just SYSOUT=A) or physical sequential data set.

The DXTDUMP data set's record length must be 121 bytes.

**DDLST, DDPUNCH, and DDINPUT**
Place your EXECUTE command for DAP in the DDINPUT data set. DAP writes the commands into DDLST, and writes the UI commands for creating the DataRefresher file and PSB descriptions into DDPUNCH.

The DDLST statement (with the parameter SYSOUT=A) routes the listing back to your system.

You must define an output data set with the DDPUNCH statement. If you want to keep your job output at the system where it originated, specify a data set name for the output in the DDPUNCH statement. Your output can be edited to add any necessary UI commands before it is submitted to the UI for processing.

You need not limit your DDINPUT data set to a single EXECUTE command for the DAP, you can add other Dictionary commands, or more EXECUTE commands for the DAP. You can do this to identify more PSBs and files than fits on a PARM= line (200 character limit).

**Note:** If you add other Dictionary commands, some may add records that do not carry UI commands to the DDPUNCH file.

**Using the DAP from the Administrative Dialogs**
To use the DAP with the Administrative Dialogs:

1. Select Option 2 Description, from the 'Administrative Dialogs' main menu.

2. Select Option 6 from the Build and 'Maintain Data Description Request' panel to invoke the DAP.

The 'DAP – Identify Data Descriptions' panel, shown in Figure 46, is displayed.

```
COMMAND ====>

Select the type of Data Dictionary
subject for which DataRefresher source form
data descriptions are to be built  ====> 2
1  FILE
2  PSB

Do you want to include subfields?
Enter Yes or No.  ====> yes

Specify the debug level (1 to 4)  ====> 1

Press:  HELP key for information  ENTER to continue  END key to return
```

Figure 46. DAP Identify Data Descriptions panel

3. Type the following information on this panel:
   - Whether a data description for a DXTFILE (option 1 File) or a DXTPSB (option 2 PSB) is to be built.
   - Whether the DAP is to define subfields.
• What debug level the DAP is to use. For more information about debug levels, see DataRefresher Command Reference.

4. Press ENTER.

The 'DAP - Data Dictionary Subjects' panel, shown in Figure 47, is displayed.

```
COMMAND ===>
DAP - DATA DICTIONARY SUBJECTS
SCROLL ===> List of PSB subjects for which data descriptions are to be built:
Note: If you specify a PSB subject occurrence number, it must be zero.

Type the appropriate letter in the SEL column on a given line and press ENTER.

i = insert one or more lines     d = delete line from list
 g = go ahead with dialog (on any line)

Press: ENTER to select update  END key to return  HELP key for information
Useful Commands:  END  HELP  UP  DOWN  CANCEL  KEYS

SEL  STATUS CODE  USER NAME  OCCURRENCE NUMBER

P  SAMPPSB1  0

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

** Figure 47. DAP Data Dictionary Subjects panel **
```

5. Type the following information on this panel:

• Status code
• User name
• Number of occurrences of the target data for the definition.

**Note:** If you specify a PSB subject occurrence number it must be zero (0).

For more information about the entries for this panel, press HELP.

6. Press ENTER.

The 'JCL Review – Data Description Request' panel, shown in Figure 26 on page 76, is displayed.

7. Select either the JCL model, DVREDDAP, or JCL based on this model, by entering the name of the JCL in the JCL file name field.

After you press ENTER, the JCL model with the DataRefresher-created EXECUTE command is displayed, so that you can check the JCL.

8. Press END, to exit from the file.

The 'Send Confirmation' panel, shown in Figure 48 on page 112, is displayed.
9. Type Y and press ENTER to send the JCL for execution.

   Note: After pressing ENTER, you may have to press the PA2 or CLEAR key to complete the confirmation.

Step 2. Editing the DAP output

Before you submit the data descriptions to the UIM, you should check the output from the DAP for any:

- Unwanted information
  
  The DAP creates UIM commands for your designated files and PSBs using all the Data Dictionary information available. As a result, the output may describe more segments, fields, or PCBs than you require.

- Missing information
  
  The subject and relational data in the Data Dictionary may be incomplete because the information about a subject can be entered incrementally. Information may be missing from the Data Dictionary because it is not normally carried in the Dictionary databases. An example is the DDNAME for a file. The DAP indicates in the DDPUNCH and DDLIST output that information is missing from the Data Dictionary.

- Incorrect information
  
  Information from the DAP could be incorrect because the Data Dictionary was created, or maintained incorrectly, or may include nonstandard representation of data.

DDPUNCH output

The DDPUNCH output contains the UIM commands that create the files and PSBs you identified in DAP input. The DAP flags the missing or invalid information in a statement by substituting asterisks for unknown values in the statement keywords. The asterisks show that the length of the segment being described is unavailable.

SEGMENT NAME=ROOT,PARENT=0,FORMAT=F,BYTES=*****
DDLIST output

The DDLIST output is part of the printout that the DAP produces. It shows the statements exactly as they appear in DDPUNCH, including asterisks for missing data. Also each line that contains missing data is flagged by three asterisks to the left of the line.

The printout can also contain the following types of DAP messages:

- Informational
- Warning
- Nonfatal error
- System error

For more information about DAP messages, see DataRefresher Messages and Codes.

Notes:

1. If a PCB uses an index field, the DAP may not generate correct FIELD statements for the root segments of that PCB. The DAP puts the SEQFLD keyword on the physical sequence field of the root segment. However, the DAP does not generate a FIELD statement if the source and target segments are different. If the DAP finds a PCB with a PROCSEQ attribute, it issues a message stating that you may need to update the FIELD statements for the root segment.

2. If the DAP finds an alternative (TP) PCB in the Data Dictionary, it automatically generates a dummy DXTPCB statement, and SEGMENT and FIELD statements for the TP PCB. The DAP also issues a message warning against using the alternative PCB when creating DataRefresher views.

3. When the DAP finds a GSAM PCB, it issues a warning message, because DataRefresher does not support GSAM databases. The DAP then attempts to continue processing the PCB.

Errors not detected by the DAP

The DAP cannot always detect all missing information and errors. The following is a list of the possible omissions and errors.

Missing FIELD keywords

When identifying a DXTPSB, you cannot be certain that every field in the associated databases has been described to the Data Dictionary.

If the FIELD keyword and associated keywords for a given segment in a DXTPCB description do not cover the entire segment, check the segment.

Missing SEGMENT keywords

The SEGMENT keyword and associated keywords generated by the DAP for a PCB might not include keywords for every segment that is visible through the PCB.

Through the Data Dictionary, the DAP can detect a missing SEGMENT keyword when it discovers a child missing a parent segment. Suppose, for example, that segments named A, B, and C are visible through a certain PCB. If the PCB describes B as a child of A, and C as a child of B, but only segment C is described in the Data Dictionary as being visible, the Data Dictionary only sees C.
Figure 49. Segment Structure

**Note:** DAP detects the absence of a description for segment B, but does not detect the absence of a description for segment A, unless A has another child that is both visible to the PCB and described as such in the Data Dictionary.

**Missing PCB keywords**

PCB keywords could be missing from the Data Dictionary. A comment containing the name qualifiers for the Dictionary subject that represents the corresponding PCB is shown to the right of each PCB keyword in the DDLIST printout. If the occurrence numbers for the PCB keywords are not consecutive integers beginning with 1, a PCB keyword could be missing.

**Missing EXIT and XBYTES keywords**

The DAP does not generate EXIT and XBYTES keywords for UIM commands to create data descriptions. You must edit DAP-generated output to add the values of those keywords, if the source data described will be changed or interpreted by a data exit.

**Incorrect data types**

Numeric fields are sometimes represented to IMS as character fields (type C). Check DAP-generated type C FIELD keywords to ensure they are correctly described.
Chapter 12. Maintaining data descriptions

This chapter describes how to delete, punch, and print the data descriptions which you have created and stored in the FDTLIB.

Deleting a data description

This section describes how to delete (remove) a data description from the FDTLIB.

To remove a data description from the FDTLIB:

1. Select option 2, Description, from the 'Administrative Dialogs' main menu, after starting a Administrative Dialogs session.
   The 'Build and Maintain Data Description Requests' panel is displayed.
2. Select option 4, Delete Data Description(s), from the FDTLIB.
   The 'Delete – Identify Data Descriptions' panel, shown in Figure 50, is displayed.

```
COMMAND ===> DELETE - IDENTIFY DATA DESCRIPTIONS

Select the type of data description to be deleted ===> 1
1 = DXTFILE
2 = DXTPSB
3 = DXTVIEW
4 = DXTRNODE
5 = DATA TYPE

If the DXTFILE or DXTVIEW data description is on a remote node, enter DXTRNODE name ===> 

Note: Remote nodes are no longer supported.

Press: HELP key for information ENTER to continue END key to return
```

Figure 50. Delete - Identify Data Descriptions panel

3. Select the type of data description to be deleted, and press ENTER.
   The 'Delete – Data Descriptions' panel, shown in Figure 51 on page 116, is displayed.
List of DXTFILE data descriptions to be DELETED:

Type the appropriate letter in the SEL column on a given line and press ENTER.

i = insert one or more lines
d = delete line from list
g = go ahead with dialog (on any line)

Press: HELP key for information ENTER to continue END key to return

SEL DATA DESCRIPTION NAME

Figure 51. Delete – Data Descriptions panel

4. Type the name or names of the DXTFILEs you want to delete.

5. Type 6 in the SEL column, next to one of the DXTFILE names, and press ENTER.

The 'JCL Review – Data Description Request' panel, shown in Figure 52, is displayed.

Figure 52. JCL Review – Data Description Request panel

6. Type the name of the JCL file that you want the Administrative Dialogs to concatenate with the delete request, and press ENTER.

An ISPF edit session is started on the job stream containing the JCL file and the DataRefresher DELETE command. For details about this command, see the DataRefresher Command Reference.

When you have completed your editing session, press END. The 'Send Confirmation' panel, shown in Figure 53 on page 117, is displayed.
7. Confirm that you want to send the delete request by typing Y in the input field, and pressing ENTER.

   **Note:** After pressing ENTER, you may have to press the PA2 or CLEAR key to complete the confirmation.

   The delete request is sent to be processed.

   Check the job output to ensure that the job completed successfully. A return code of zero (0) means that the job has been completed successfully.

---

**Printing a data description**

This section describes how to print and punch data descriptions:

**Printing a data description**

Retrieves a copy of a data description as it appears in the FDTLIB, and appends the data description to the DXTPRINT data set.

**Punching**

Retrieves a copy of the description as it appears in the FDTLIB, and appends the data description to the DXTPUNCH data set.

To print or punch a data description:

1. Select option 2, Description, from the 'Administrative Dialogs' main menu, after starting a Administrative Dialogs session.

   The 'Build and Maintain Data Description Requests' panel, shown in Figure 54 on page 118, is displayed.
BUILD AND MAINTAIN DATA DESCRIPTION REQUESTS

SELECT OPTION ===> 3

1  EDIT, CREATE, ERASE or SEND a data description request.
2  IMPORT or EXPORT a data description request.
3  PRINT or PUNCH data description(s).
4  DELETE data description(s) from the FDTLIB.
5  Invoke the SAP (Structure Access Program).

If you have the appropriate Program(s) installed:
6  Invoke the DAP (Dictionary Access Program).

Figure 54. Build and Maintain Data Description Requests panel

2. Select option 3, Print or Punch data description(s).

The 'Print/Punch – Identify Data Descriptions' panel, shown in Figure 55, is displayed.

PRINT/PUNCH - IDENTIFY DATA DESCRIPTIONS

COMMAND ===> 

Select the type of output for the data descriptions ===> 1
1 = PRINT
2 = PUNCH

Select the type of data description ===> 1
1 = DXTFILE
2 = DXTPSB
3 = DXTPCB
4 = DXVIEW
5 = DXRNODE
6 = DATA TYPE

If you selected a DXTPCB type above, enter the name of the DXTPSB ===> 

If the DXVIEW or DXTFILE data description is on a remote node, enter DXRNODE name ===> 

Press:  HELP key for information  ENTER to continue  END key to return

Note: Remote nodes are no longer supported.

Figure 55. Print/Punch – Identify Data Descriptions panel

3. This panel prompts you to select:
   
   • Whether you want to print or punch the data description.
   
   • Select the type of data description.

   If you select a PCB, you must identify the DXTPSB which contains the PCB.

   Type your selections and press ENTER. The 'Print/Punch – Data Descriptions' panel, shown in Figure 56 on page 119 is displayed
4. The panel prompts you type the name of the DXFILE, DXTPSB, DXTVIEW, or DXTPCB you want to print. You can enter up to 16 names on the panel, or specify an asterisk (*) to print all data descriptions.

5. After typing the file name or names, type G in the SEL column and press ENTER.

The 'JCL review – Data Description Request' panel, shown in Figure 57 is displayed.

6. Type the name of the JCL file that you want concatenated with the print or punch request.

   This file invokes UIM to retrieve a copy of the data description from the FDTLIB.

7. Press ENTER.

   An ISPF edit session is started on the job stream containing the JCL file and DataRefresher PRINT or PUNCH command. For information about the commands, see the DataRefresher Command Reference.

   When you have completed the edit session, press END (F3). The 'Send Confirmation' panel, shown in Figure 58 on page 120, is displayed.
Figure 58. Send Confirmation panel

8. Type Y in the input field, if you want to confirm that the request is to be sent to the FDTLIB, and press ENTER.

Note: Check the output from your job to see whether the job completed successfully. If it completed successfully, a return code of 0 is returned.
Chapter 13. Maintaining extract requests

The STATUS, CANCEL, and LIST DataRefresher commands are available for maintaining DEM extract requests:

- Using STATUS in the Administrative Dialogs lets you check the status of an extract request in the EXTLIB, to see if it has been processed.
- Using CANCEL in the Administrative Dialogs lets you delete an extract request.
- Using LIST in the Administrative Dialogs lets you list all, or some, of the extract requests currently waiting for processing in the EXTLIB.

For more information about these commands and their syntax, see the DataRefresher Command Reference.

Checking the status of an extract request

To check the status of an extract request:

1. Select option 1 from the 'Administrative Dialogs' main menu to display the 'Build and Maintain Extract Requests' panel, shown in Figure 59, after starting an Administrative Dialogs session.

   ![](image)

**Figure 59. Build and Maintain Extract Requests panel**

2. Select option 3 from the 'Build and Maintain Extract Requests' panel, to check the status of an extract request.

   The 'Status – Identify Extract ID' panel, shown in Figure 60 on page 122, is displayed.
STATUS - IDENTIFY EXTRACT ID

COMMAND ===>

Fill in the following information:

Extract id ===>
User id (optional) ===>
Node id (optional) ===>

Figure 60. Status – Identify Extract ID panel

The 'Status – Identify Extract ID' panel prompts you for the EXTDI keyword value in the SUBMIT command of the extract request, your user ID, and the node of the system you are working on. The user ID and node ID fields are optional.

3. Type your entries on the panel, and press ENTER.

The 'JCL Review - Extract Request' panel, shown in Figure 61, is displayed. This panel prompts you for the name of the JCL that invokes the UIM, to process the STATUS command, in the 'JCL Review - Extract Request' panel. The load JCS file name and JCS DD name entries are optional.

JCL REVIEW - EXTRACT REQUEST

COMMAND ===>

This is the JCL name associated with the request:
You may change the following names for this request WITHOUT affecting the defaults established in your profile.

Execution JCL ===>
Load JCS ===>
JCS DDname ===>

To display a list of existing JCL files, enter a ? in the JCL file name field.

If you are sending a request, after pressing the ENTER key you may have to press the PA2 or CLEAR key to complete the dialog.

Press: HELP key for information ENTER to continue END key to return

Figure 61. JCL Review – Extract Request panel

4. Type your entries on the panel, and press ENTER.

5. An ISPF edit session is started on the job stream containing the JCL and status command. For details about the syntax of the STATUS command, see DataRefresher Command Reference.

After you finish creating your request, press END.

6. The 'Send Confirmation' panel is displayed. Type Y in the input field and press ENTER to confirm that the command is to be sent.

Note: After pressing ENTER, you may have to press the PA2 or CLEAR key to complete the confirmation.
When the confirmation has been sent, and the JCL processed, the dialogs will return to the 'Build and Maintain Extract Requests' panel. To check the status of the extract request look in the DXTPRINT file produced by the UIM job.

Deleting an extract request from an EXTLIB

To delete an extract request:

1. Select option 1, Extract, from the 'Administrative Dialogs' main menu, to display the 'Build and Maintain Extract Request' panel, after starting an Administrative Dialogs session.

2. Select option 4, Cancel, from the 'Build and Maintain Extract Request' panel.

The 'Cancel - Identify Extract ID' panel, shown in Figure 62, is displayed.

```
COMMAND ===> CANCEL - IDENTIFY EXTRACT ID

Fill in the following information:

Extract id ===> 

User id (optional) ===> 

Node id (optional) ===> 

After pressing the ENTER key to complete and send the CANCEL command, you may have to press the PA2 or CLEAR key to complete the dialog.
```

Figure 62. Cancel - Identify Extract ID panel

3. Type the Extract ID used in the EXTID keyword value of the SUBMIT command for the extract request you want to cancel, and press ENTER. The User ID and Node ID fields are optional.

   **Note:** After pressing ENTER, you may have to press the PA2 or CLEAR key to complete the confirmation.

The 'JCL Review - Extract Request' panel, shown in Figure 61 on page 122, is displayed.

4. Type the name of the JCL file that invokes the UIM to process the CANCEL command in the Execution JCL field, and press ENTER. The JCS file name and JCS DD name fields are optional.

   **Note:** After pressing ENTER, you may have to press the PA2 or CLEAR key to complete the confirmation.

5. An ISPF edit session is started on the job stream containing the JCL and CANCEL command. For details about the syntax of the CANCEL command, see *DataRefresher Command Reference*.

   When you have created your command, press END to complete the ISPF edit session. The 'Send Confirmation' panel is then displayed.

6. Type Y in the input field and press ENTER to confirm that the command is to be sent.

   **Note:** After pressing ENTER, you may have to press the PA2 or CLEAR key to complete the confirmation.
When the confirmation has been sent, and the JCL processed, the extract request is deleted from the EXTLIB.

Listing extract requests in an EXTLIB

To list the extract requests contained in the EXTLIB:

1. Select option 1, Extract, from the 'Administrative Dialogs' main menu after starting an Administrative Dialogs session.

2. Select option 5, List, from the 'Build and Maintain Extract Requests' panel.

The 'List – Identify Extract Request Parameters' panel, shown in Figure 63, is displayed.

3. Specify the parameters by which the extract requests in the EXTLIB are to be listed, and press ENTER.

In the example shown below the only parameter is an A in the State input field. This specifies that all extract requests in the EXTLIB are to be listed. The other fields are optional and can be ignored.

**Note:** After pressing ENTER, you may have to press the PA2 or CLEAR key to complete the confirmation.

```
LIST - IDENTIFY EXTRACT REQUEST PARAMETERS
COMMAND ===> 
Fill in the following information:
User ID
(optional) ===> 
Node ID
(optional) ===> 
State
   Enter Any or Waiting. ===> A

If you specify a * in the User ID or Node ID fields, then
the extract requests for all users or nodes are LISTed.

After pressing the ENTER key to complete and send the LIST request,
you may have to press the PA2 or CLEAR key to complete the dialog.
Press: HELP key for information ENTER to continue END key to return
```

*Figure 63. List - Identify Extract Request Parameters panel*

The 'JCL Review - Extract Request' panel, shown in Figure 61 on page 122, is displayed.

4. Type the name of JCL that invokes the UIM to process the LIST command, and press ENTER. The load JCS file name and JCS DD name are optional.

**Note:** After pressing ENTER, you may have to press the PA2 or CLEAR key to complete the confirmation.

5. An ISPF edit session is started on the job stream containing the JCL and LIST command. For more information about the syntax of the LIST commands, see *DataRefresher Command Reference*.

When you have completed the command, press END to end the ISPF edit session.
6. The 'Send Confirmation' panel is displayed. Type Y in the input field and press ENTER to confirm that the command is to be sent.

**Note:** After pressing ENTER, you may have to press the PA2 or CLEAR key to complete the confirmation.

When the confirmation has been sent, and the JCL processed, the dialogs will return to the 'Build and Maintain Extract Requests' panel. The list of extract requests will be contained in the DXTPRINT file produced by the UIM job.
Chapter 14. Changing the default profile

When you run the DataRefresher dialogs run for the first time, a default profile is created. This profile is used to:

- Control what the DataRefresher dialogs do when you build, update, or send data description requests or extract requests.
- Maintain the control information for DEM and REM.
- Control what JCL or JCS the End User Dialogs use to build a job stream and send a request.

The DataRefresher administrator can change the default profile to meet your site's requirement. You can make further changes to your own profiles using the Administrative Dialogs. End users can also change their profile by completing the 'Profile Review' panel using the End User Dialogs. For information about the 'Profile Review' panel, see Chapter 21, "JCL/JCS files and End User Dialogs requests" on page 193.

To change the default profile, you must start an Administrative Dialogs session, and select option 4, Profile, from the Administrative Dialogs main menu. A series of seven panels are displayed, on which you enter the information for the profile. These panels are described in the sections below.

Dialogs Profile Options panel 1

The fields on this panel specify which JCL file should be used for each request. To display help for the fields, press F1 (HELP). The appropriate JCL file is associated with a request through these profile fields or by using a nickname table.

```
COMMAND  ===> DIALOGS PROFILE OPTIONS  Panel 1 of 7

Specify default names to be used by all DataRefresher Dialogs:

UIM JCL  ===> DVREDXT  Enter JCL file name to invoke the UIM
DAP JCL  ===> DVREDDAP  Enter JCL file name to invoke the DAP
REM JCL (MVS) ===> DVREDREM  Enter JCL file name to invoke the REM on an MVS system.
REM JCL (VM) ===> DVREDREV  Enter JCL file name to invoke the REM on a VM system.

Enter a ? in any field to display a list of names.
Press:  HELP key for information  ENTER to continue  END key to return
```

Figure 64. Dialogs Profile Options panel 1

If this is the first time you have displayed this panel, the JCL fields contain the names of the model JCL files shipped with DataRefresher dialogs. You need to change these fields, if you want different JCL files associated with your extract requests.
To display a list of JCL file names, place your cursor in a field, type a question mark (?), and press ENTER. The 'List of JCL File Names' panel is displayed.

To go to the next panel in the series, press ENTER.

**Dialogs Profile Options panel 2**

The fields on this panel specify which JCS file DataRefresher associates with various End User Dialogs extract requests. DataRefresher associates the appropriate JCS file with a request based on the target type field for the request. For example, if the target type is DB2, the DB2 JCS specified on this panel is used.

![Diagram](image)

*Figure 65. Dialogs Profile Options panel 2*

If this is the first time you have displayed this panel, the default JCS models that are shipped with DataRefresher dialogs are used in these fields. You need to change these fields, if you want to have different JCS

If you want to see a list of possible JCS files you can specify, place your cursor in a field, type a question mark (?), and press ENTER.

The three fields, Message JCS, Message table, and User table qualifier, identify the message table, which is a relational database table set up to receive DataRefresher messages. You should only make an entry in these fields if the UIM messages are to be loaded into a message table. See the DataRefresher Administration Guide for information about the format that a message table must have to receive UIM messages. If no relational table is to be used for DataRefresher messages, do not make an entry in the fields.

To go to the next panel in the series, press ENTER.
Dialogs Profile Options panel 3

This fields on this panel are used to specify the SUBMIT and EXTRACT options for End User Dialogs requests.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output limit</td>
<td>Specify 1 to 10000000 or None</td>
</tr>
<tr>
<td>Priority</td>
<td>Specify 0 to 255</td>
</tr>
<tr>
<td>Debug level</td>
<td>Specify 1 to 4</td>
</tr>
<tr>
<td>Field error</td>
<td>Specify field error action</td>
</tr>
<tr>
<td>Message limit</td>
<td>Specify 0 to 1000000 to limit the field error messages</td>
</tr>
<tr>
<td>Decimal Point</td>
<td>Specify 1 = Period, 2 = Comma</td>
</tr>
<tr>
<td>Format</td>
<td>Specify the format of extracted data</td>
</tr>
<tr>
<td></td>
<td>1 EBCDIC converted format</td>
</tr>
<tr>
<td></td>
<td>2 SOURCE format</td>
</tr>
<tr>
<td>System ID</td>
<td>Specify your system Node or nickname</td>
</tr>
</tbody>
</table>

Press: HELP key for information ENTER to continue END key to return

Figure 66. Dialogs Profile Options panel 3

The fields on this panel are:

**Output limit** Controls the number of rows of output that DEM or REM produces and lets your site schedule when your extract requests run according to size. For information about what to specify on the panel, press F1 (HELP).

**Priority** Determines when the DEM runs your extract request. The higher your priority value, the quicker your request is executed. For a detailed explanation of priorities, see the DataRefresher Command Reference.

**Debug level** Determines the level of diagnostic information DEM maintains and writes when it runs your extract requests.

Table 11 shows the diagnostic levels for DEM.
Table 11. Diagnostic information maintained by DEM per debug level

<table>
<thead>
<tr>
<th>Type of Diagnostic Information</th>
<th>LEVEL 1</th>
<th>LEVEL 2</th>
<th>LEVEL 3</th>
<th>LEVEL 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEM maintains an internal wrap-around trace table showing module entry and exit points</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DEM keeps trace information about all data access method and source data set (IMS/VS DL/I, physical sequential, and VSAM) calls</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DEM prints trace data whenever it enters or exits a module or a user exit routine</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DEM prints control area information whenever it enters or exits a module, or other internally specified locations and includes dumps of user exit routine control blocks</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

**Note:** The higher the debug level, the more the impact on DEM performance when it processes your extract requests.

**Field error**

Controls what DEM does if it meets an incorrectly formatted field when running your extract requests. Do not type imbedded blanks in a Field error value. For information about what to specify, press F1 (HELP).

If an extract request is stopped because of a field error, the DEM writes all of the output up to the point of that field error and removes the extract request from the EXTLIB.

**Message limit**

Controls the number of messages the DEM sends to the message file for field errors.

Use caution when setting this message limit. If it is set too low, you can limit the amount of diagnosis you can do if a request fails. The unexpected messages may occur past the limit you have set for the particular request.

If you set a very high limit and there are many field error messages, this will result in a large message file.

**Decimal Point**

Determines what character DataRefresher uses as the decimal place marker in the extract output. Set this to:

1. For a period indicator
2. For a comma indicator

**Format**

Determines whether your extracted data is be output in EBCDIC or SOURCE format from the DEM or REM.

**System ID**

Identifies the system on which the extract request resides. System ID can be any alphanumeric name, and is used by both DataRefresher End User Dialogs and DataRefresher Administrative Dialogs.

While you can type any alphanumeric name in this field, it is suggested that you use your node ID or the node nickname assigned by your site. This ensures that your extract request is
uniquely identified and your DXTPRINT output is routed back to you.

To go to the next panel in the series, press ENTER.

**Dialogs Profile Options panel 4**

The fields on this panel specify additional SUBMIT command options for DataRefresher End User Dialogs requests.

<table>
<thead>
<tr>
<th>Command</th>
<th>Specify the SUBMIT OPTIONS for End User Dialogs (quotes optional):</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCOUNT</td>
<td>Specify accounting information</td>
</tr>
<tr>
<td>USERDECK</td>
<td>Override options on DB2 load control deck</td>
</tr>
<tr>
<td>USERDECK</td>
<td>Override options on SQL/DS load control deck</td>
</tr>
<tr>
<td>MAP EXIT</td>
<td>Specify a Map Capture Exit and optional parameters.</td>
</tr>
<tr>
<td>EXIT PARMS</td>
<td>Press: HELP key for information ENTER to continue END key to return</td>
</tr>
</tbody>
</table>

*Figure 67. Dialogs Profile Options panel 4*

The fields on this panel are:

**Account** Specifies your accounting information. It is passed to the accounting exit routine set up to monitor DEM activity at your site.

**Userdeck (DB2)** Specifies the DB2 LOAD options. The commands you specify now override the default LOAD statement of the control deck that is generated by the DB2 load utility when processing your End User Dialogs request. For the information about the LOAD statement, see *IBM DATABASE 2 Version 2: Command and Utility Reference*.

**Userdeck (SQL/DS)** Specifies the options to be included in the DATALOAD input. The statements you specify in this field override the default INFILE statement of the control deck that is generated by the SQL/DS DATALOAD utility when processing your End User Dialogs request. For the options on the DATALOAD statement in SQL/DS, see *SQL/Data System Database Services Utility for IBM VM Systems*.

**MAP EXIT** Specifies the user-written map capture exit routine that is invoked when DEM or REM processes your extract request. This parameter does not apply to REM extract requests. For more information about Map Capture exit routines, refer to *DataRefresher Exit Routines* book.
EXIT PARMS

Specifies the parameters that are passed to the map capture exit routines.

For more information about the fields, press HELP (F1). To go to the next panel in the series, press ENTER.

Dialogs Profile Options panel 5

The fields on this panel specify the extract output options for DataRefresher End User Dialogs.

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DIALOGS PROFILE OPTIONS</th>
<th>Panel 5 of 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the EXTRACT OUTPUT OPTIONS for End User Dialogs:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control Deck</td>
<td>➔ 1</td>
<td>Select the LOAD Control Deck routing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Include with JCS data set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 Write to physical sequential data set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 Do not generate a control deck</td>
</tr>
<tr>
<td>Extract Output</td>
<td>➔ 1</td>
<td>Select the routing of extract output</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Generate a job using the JCS data set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 Write to physical sequential data set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 Perform both of the above</td>
</tr>
<tr>
<td>Press: HELP key for information ENTER to continue END key to return</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 68. Dialogs Profile Options panel 5

The fields on this panel are:

Control Deck

Specifies whether DEM or REM is to generate a control deck, and if it is generated where it should be placed in the output. The control deck contains the statements necessary to run the DB2 or SQL/DS load utility.

Extract Output

Specifies where the extract output is routed. If you want your extract output to be immediately loaded into a relational table, you need to have the JCS required to do the load.

To go to the next panel in the series, press ENTER.
Dialogs Profile Options panel 6

The fields on this panel specify the additional SUBMIT command parameters for End User Dialogs extract requests.

```
COMMAND ===> DIALOGS PROFILE OPTIONS Panel 6 of 7
If extract output is to be routed to physical sequential data set, specify the
DDname that defines the data set:
 DDname in DEM JCL ===>  
 DDname in REM JCL (MVS) ===>  
 DDname in REM JCL (VM) ===>  
OR, Specify the attributes if dynamic allocation is preferred:
 Data Set Name ===>  
 Unit ===>  
 Volume serial ===>  
 Blocking factor ===>  
 Password ===>  
 CMS File name ===>  
 File type ===>  
 File mode ===>  
Press: HELP key for information ENTER to continue END key to return
```

Figure 69. Dialogs Profile Options panel 6

The panel displays the fields that let you name a physical sequential data set in
which you want your extract output written. For the meaning of each field, press F1
(HELP).

The information specified on this panel is substituted on the EXTDATA keyword of
the SUBMIT statement when the End User Dialogs builds extract requests.

The entries you make depend on the operating system:

**MVS** If you want the data set dynamically allocated, enter the physical attributes
of the output data set. If you do not want dynamic allocation, specify the
DD name in the DEM or REM execution JCL. When you specify both, the
DEM or REM DD name is used.

**VM** If you want the output file dynamically allocated, enter the full file name of
the output file. If you do not want dynamic allocation, specify the DD name
in the REM execution JCL. When you specify both, the REM DD name is
used.

To go to the next panel in the series, press ENTER.
Dialogs Profile Options panel 7

This panel displays the processing options that the Administrative Dialogs use:

COMMAND ===> DIALOGS PROFILE OPTIONS Panel 7 of 7

Select the Administrative Dialogs PROCESSING OPTIONS you prefer:

Enter Yes or No for each of the following:

JCL Review ===> Y Enter 'yes' to review the JCL names each time you build or send an extract request.

Job Edit ===> Y Enter 'yes' to edit the generated job stream each time you send a request.

Press:  HELP key for information  ENTER to continue  END key to return

Figure 70. Dialogs Profile Options panel 7

The fields on this panel are:

JCL Review
Determines whether you want to review the JCL names each time you build, update, or send an extract request or data description request.

If you enter a Y in this field, a panel displaying the default JCL and JCS associated with your extract request or data description request is displayed. You can change the JCS and JCS file names that you want to have associated with the request by replacing the default values on this panel.

Job Edit
Determines whether you want to edit the generated job stream before a request is sent for execution. This option also lets you edit PRINT, PUNCH, DELETE, DAP, STATUS, CANCEL, REQUEST SOURCE DESCRIPTION, and LIST requests.

For information about these requests, see the DataRefresher Command Reference.

If you have made all the changes you need to the profile, press the ENTER key. This saves the changes and returns you to the Administrative Dialogs main menu.
Chapter 15. Providing access to dialog objects

This section describes the different ways that user access can be provided to DataRefresher dialogs objects, such as data descriptions, extract requests, and JCL/JCS files. There are three ways to provide access to the files:

- Provide each user with edited copies of the files
- Use object sharing
- Use object sharing and maintain personal DataRefresher libraries.

You must select how users can access DataRefresher objects at your site. For example, if you have users who occasionally run extract requests, they should use object sharing. This lets them use extract requests that you maintain in a central library.

Alternatively, if you have several users who create and save extract requests in their own DataRefresher libraries, you can set up object sharing for them, and let them have their own DataRefresher libraries. This lets them use shared objects as models, then change and save them in their own individual libraries.

Providing edited JCL files

This section describes the tasks you must complete if each user maintains a personal DataRefresher dialogs library:

Notes:

1. If you edit the models, and do not create new files with new names, your users automatically receive the edited models when you enroll them.

   If you create files with different names, users will not automatically receive the files.

2. If you create a dialog object (extract, data description, or JCL file) that is to be shared, you can export the object into the DVRIMEXE data set and provide access to this data set for other users. The other users can then import the object from your DVRIMEXE into their own data libraries.

1. Log on to the Administrative Dialogs.

2. From the Administrative Dialogs main menu, select option 3, JCL/JCS.

   The 'Build and Maintain JCL Files' panel, shown in Figure 71, is displayed.

   ![Figure 71. Build and Maintain JCL Files panel](image)

   **Figure 71. Build and Maintain JCL Files panel**

3. Select option 2, Import or Export, to export the file.

   The 'Select Import/Export Function' panel, shown in Figure 72 on page 136 is displayed:
SELECT OPTION ===> 2

1  IMPORT a DataRefresher Dialogs object
2  EXPORT a DataRefresher Dialogs object
3  ERASE object(s) from Import/Export library

Enter the name of the dataset that you want to import from or export to. This must be an existing dataset. Enclose the name in single quotes if you do not want a prefix inserted.

DATASET NAME: DVR110.DVRIMEXE

Figure 72. Select Import/Export Function panel

4. Select option 2, Export, to export your objects.

The default Import/Export library name is

- DVR110.DVRIMEXE, in MVS
- DVRIMEXE MACLIB, in VM

The 'List of JCL Files Names' panel, shown in Figure 73, is displayed. This panel displays a list of all the JCL files in your dialog library. The following is an example of this panel.

Figure 73. List of JCL File Names panel

5. Type an S next to the file you want to export. You may only export one file at a time.

6. Press ENTER.

The 'Identify JCL/JCS Name' panel, shown in Figure 74 on page 137, is displayed.
SELECT OPTION ===>  

Create and enter a NAME for the JCL/JCS you are exporting.  
Enter a ? in NAME field to display a list of existing objects.  
The DESCRIPTION field is optional but highly recommended.

NAME ===>  
DESCRIPTION ===>  

Figure 74. Select Import/Export Function panel

7. Type the name of the file, and a description of the file in the Identify JCL/JCS  
Name panel.

8. Press ENTER.  
The file is stored in the Import/Export library, and the 'Select Import/Export  
Function' panel is redisplayed with a message telling you that the file was  
stored successfully.

The new users can use the Import dialog option, from their own user IDs, to receive  
the JCL models. For instructions on how to import a dialog, see “Enrolling MVS  
DataRefresher dialogs users” on page 149.

Using object sharing

Object sharing is a method of obtaining an object or a list of objects from common  
(shared) disks or data sets, for example, a central library. If you want to maintain a  
central DataRefresher library, set up your DataRefresher dialogs environment for  
object sharing. Object sharing can be used with End User Dialogs and  
Administrative Dialogs. It lets all users with access to the shared library access  
data descriptions, extract requests, and JCL/JCS files.

Object sharing must be initiated with each DataRefresher session, the logon CLIST,  
DVRELLGN, or the logon EXEC, DVREXUSR. The shared objects are available on  
the data set or file allocated to the DD name DVRSTABL.

The coding example depends on the operating system. For:

**MVS** You would code:  

```
LIBDEF DVRSTABL DATASET ID('DVR110.DVRTLIBE')
```

**VM** You would code:  

```
LIBDEF DVRSTABL FILE ID(DVRTLIBE MACLIB x)
```
Where x is the file mode of a shared minidisk.

**Note:** It is recommended that the shared minidisk is the production disk.

Security on the shared library is provided by your normal data set or file  
procedures:

**MVS** The shared library should be protected, so that only the database  
administrator is permitted to update the library.

**VM** The shared library should be in read-only mode, except when you are  
updating the library.
If other users need to update certain data descriptions or extract requests, they create the object locally as their own data set, and send the object to you. You can import the object to the shared library. The process is similar if another user creates a new JCL file or extract request that has to be placed in the shared library.

**Note:** DataRefresher stores the password (provided on the job card) in the DVRJEDIE data set or MACLIB. Care should be taken when using object sharing capability to avoid a security exposure at your site.

All DataRefresher End User Dialogs commands are read-only during an object sharing session, except for ERASE and SAVE. You cannot issue an ERASE command during a shared objects session unless you have write access to the shared objects library. You can, however, issue a SAVE command. If you are not authorized to update the shared objects library, DataRefresher saves the object in your own DataRefresher dialogs library.

---

**Using object sharing and maintaining a personal DataRefresher library**

You can use a combination of the methods described in the section "Providing edited JCL files" and "Using object sharing." This approach is suitable for users who have some data processing experience and are responsible for creating some extract requests. For example, if you have a user with some data processing experience who has been using the End User Dialogs to create extract requests.

You can create several complicated End User Dialogs extract requests, and store them in the shared objects library. Users can use these extract requests as models by accessing them through DataRefresher End User Dialogs, making the necessary changes and saving them in their personal DataRefresher libraries.

**Note:** The individual extract requests are not available in the extract list in the DataRefresher End User Dialogs shared session until the user restarts the unshared session by returning to the ISPF main menu and re-accessing DataRefresher End User Dialogs.
Chapter 16. Setting up End User Dialogs

This chapter describes the following DataRefresher administration tasks which need to be performed before users can access the End User Dialogs:

- Building and updating the nickname table
  
The nickname table contains the nicknames that are assigned to specific nodes or subsystems and the JCL used to route requests to those systems. For information about setting up the nickname table, see “Building or updating the nickname table” on page 140.

- Updating the end user table
  
The end user table identifies all users who have access to End User Dialogs. For information about setting up the end user table, see “Updating the end user table” on page 141.

- Requesting source table description data
  
For information about requesting source table data, see “Requesting source table description data” on page 142.

- Providing access to the master index table
  
The master index table (MIT) is a list of DataRefresher views and tables that an End User Dialogs user is authorized to access. For information about setting up the master index table, see “Providing access to the master index table (MIT)” on page 148.

- Enroll the End User Dialog users

The nickname table, end user table (EUT), and master index table (MIT) reside in the data set allocated to the DD name DVREUADD. The default name for the data set or file depends on the operating environment:

MVS    DVR110.DVRTADME
VM     DVRTADME MACLIB

In this chapter, both defaults are called DVRTADME. Before starting these administrative tasks, ensure that you have write access to the DVRTADME data set or file.
Building or updating the nickname table

The nickname table enables you to:

- Distinguish between subsystems on the same MVS or VM system. For example, a subsystem could be:
  - A DB2 subsystem
  - An SQL/DS subsystem
  - A set of FDTLIB and EXTLIB libraries

It is possible to use the same table or DataRefresher view name, in different FDTLIBs or system catalogs.

**Note:** If two DataRefresher views or tables exist with the same name, but in different MVS systems, then a nickname table is not required, because the End User Dialogs panel, 'Select Tables/DataRefresher Views for Extract', specifies the location of each view. An end user can select the correct DataRefresher view by choosing the appropriate location.

- Specify an alternate name for a system.

A nickname table lets you to create a naming convention for the object list, and lets end users refer to the same objects with a significant name.

You should use nicknames to specify JCL or JCS files that are used with a particular system.

To build or update the nickname table follow the steps below:

1. Select option 5, Administer, from the Administrative Dialogs main menu, after starting an Administrative Dialogs session.
   The 'End User Dialogs Administration' panel is displayed.

2. Select option 1, Build/Update nickname table, from the 'End User Dialogs Administration' panel.
   The 'Establish Nicknames For Systems' panel is displayed.

```
COMAND <--- ESTABLISH NICKNAMES FOR SYSTEMS ROW 1 OF 1 SCROLL <--- PAGE

Enter information about the systems, both source and target, that you are accessing. All fields may not apply to each system. The Nickname you create will be referred to throughout the Dialogs.

Press: HELP key for information ENTER to continue END key to return

SEL : i = INSERT one or more lines, d = DELETE line from list

SEL NICKNAME NODE ID JCL/JCS DATA TYPE LOCATION DESCRIPTION
FILE NAME 1=DB2,2=SQLDS, 1=SOURCE,
(? for list) 3=DVR, 2=TARGET
4=IXF,5=SELECT
```

*Figure 75. Establish Nicknames for Systems panel*

3. Type your nickname information in the appropriate fields. For help about the contents of each field on the panel, press HELP (F1).

4. Press END key when you have completed the panel.
Updating the end user table

The end user table (EUT) enables end users to access source data which could reside on either the same or a different system as the user. The table acts as a cross reference table and resides in a data set named DVRTADME.

The EUT is an ISPF table which consists of the following columns:

- The node ID or nickname of the system containing the source data
- The user ID on that node or nickname
- The End User Dialogs user ID

The End User Dialogs user ID column is the Dialogs user ID from which extract requests are built. End users may have different user IDs on different systems.

Depending on your local security and authorization, you may not have to update the EUT at all. The EUT is automatically created with the following entries:

```
SOURCE NODE = *
USERID AT SOURCE = PUBLIC
```

This lets all users extract public data on all node IDs. You cannot delete this entry from the EUT. If all the data descriptions and tables you plan to provide your end users in the next step are public, you do not need to make any changes to the EUT.

However, if the data descriptions or the tables you are providing to end users have been granted specific authorization, you must create an EUT entry for each user.

To update the EUT follow the steps shown below:

1. Select option 5, Administer, from the Administrative Dialogs main menu, after starting an Administrative Dialogs session.
   
   The 'End User Dialogs Administration' panel is displayed.

2. Select option 2, Update the end user table, from the panel.
   
   The 'Update the End User Table' panel is displayed:

   ![Update the End-User Table panel](image)

   **Figure 76. Update the End-User Table panel**

3. Type the appropriate source node ID, source user ID, and DataRefresher dialogs user ID in the fields of the panel.
Each entry in the EUT must be unique; no two rows may have the same entries. For information about the fields on the panel, press HELP (F1).

4. Press END key when you have completed your entries.

---

### Requesting source table description data

To create or update a list of tables and DataRefresher views available to end users currently defined in the end user table, use the Administrative Dialogs to request source table descriptions.

When you run a Request Source Table Descriptions dialog, you build a UIM or a REM job that:

1. Retrieves data descriptions from the file description table library (FDTLIB), or the DB2 or SQL/DS catalog.
2. Imbeds the data in an output job built from the inline JCS with DD name EUEIJCS. The output job creates or updates the master index table (MIT) in the DVRTADME.

You must provide all the DataRefresher views and tables that your end users need to access in the MIT. You may need to update the MIT at various times:

- To add a new table
- To change the structure of a current table
- To delete an old table

You have to update the MIT for each subsystem that contains data you want to extract. For example, if you have two DB2 subsystems, you should run the MIT job twice. Each request returns description information from one subsystem.

To request source description data, perform the following steps:

1. Select option 5, Administer, from the Administrative Dialogs main menu, after starting an Administrative Dialogs session.

   The 'End User Dialogs Administration' panel is displayed.

2. Select option 3, Request Source Descriptions, from this panel.

   The 'Request Source Table Description(s)' panel, is displayed.
REQUEST SOURCE TABLE DESCRIPTION(S)

Command: ===>

Please provide the following information about the tables/views:

- System where Source data is located (7 for a list) ===>
- Database Type (1 = DB2, 2 = SQL/DS, 3 = DXT) ===>
- User ID and Password used to access data (Required for DB2 and SQL/DS only)
  Database User ID ===>
  Database Password ===>
- VM execution ID (SQL/DS only) ===> CMSBATCH
- DXTNODE name (Remote Data Extract only) ===>

If requesting data from SQL/DS, also specify the information needed for the WMSPOOL parameter of the SUBMIT statement:

- DataRefresher Dialogs System ===>
- User ID on that system ===>
- Via Network ID ===>

Press: HELP key for information ENTER to continue END key to return

Figure 77. Request Source Table Description(s) panel 1

3. Type your selection into the appropriate fields of the panel, to select the source description data you want. For descriptions of the fields, press HELP (F1).

For an explanation of how to request source description data, see the following sections:

- "Editing a UIM job to build or update the master index table (MIT)" on page 145.
- "Editing a REM job to build or update the master index table (MIT)" on page 146.

4. Press ENTER, when you have completed your entries.

The 'Request Source Table Description(s)' panel is displayed.

REQUEST SOURCE TABLE DESCRIPTION(S)

Command: ===>

Name the tables/views to be described. Those named must be located on the same system.
An * (asterisk) in the first data entry field means all tables/views on that system.

===>

Figure 78. Request Source Table Description(s) panel 2

5. Type the names of the tables or DataRefresher views you want on this panel, or type an asterisk (*) to include all tables and DataRefresher views.

6. Press ENTER.
7. If the JCL Review option is set on, the 'JCL Review – End User Dialogs Administration' panel, shown below, is displayed. This option is set in your Dialog Profile.

You can override the JCL/JCS used to route the job and its output by changing the values shown on this panel.

```
COMMAND ===> JCL REVIEW - END USER DIALOOGS ADMINISTRATION

Either your profile indicates that you wish to review the JCL name associated with this End User Dialogs Administration request, or you are sending a request but have not made a JCL association.

You may change the following JCL name for this request WITHOUT affecting the defaults established in your profile.

  JCL FILE NAME ===> DVREDREM
  LOAD JCS NAME ===> DVREDREM

  To display a list of existing JCL files, enter a ? in either field.

If you are sending a request, after pressing the ENTER key you may have to press the PA2 or CLEAR key to complete the dialog.

Press: HELP key for information ENTER to continue END key to return
```

Figure 79. JCL Review - End User Dialogs Administration panel

8. Press ENTER, after completing your entries.

  **Note:** After pressing ENTER, you may have to press the PA2 or CLEAR key to complete the confirmation.

If JCL Review is switched off, DataRefresher displays the 'Send Confirmation' panel.

9. Type Y, and press ENTER to send the job to be processed.

  **Note:** After pressing ENTER, you may have to press the PA2 or CLEAR key to complete the confirmation.

After you complete the request for source description data in the Administrative Dialogs, you may run the MIT utility. The following sections describe how to review and edit your MIT job before you send it.

- **“Editing a UIM job to build or update the master index table (MIT)”** on page 145 describes the procedure for extracting DataRefresher view description data.

  Use this section to edit UIM jobs for end users who want to extract data from IMS DL/I databases, VSAM data sets, physical sequential data sets, or GDI data sources using the End User Dialogs.

- **“Editing a REM job to build or update the master index table (MIT)”** on page 146 describes the procedure for extracting table description data for relational systems.

  Use this section to edit REM jobs for end users who want to extract data from a DB2 or SQL/DS database using the End User Dialogs.
Editing a UIM job to build or update the master index table (MIT)

If the end user needs to extract data from an IMS DL/I database, VSAM data set, physical sequential data set, or GDI data source, you must run a UIM job to:

- Request the appropriate existing DataRefresher view descriptions from the FDTLIB.
- Run the MIT utility to update the MIT for the end user.

1. In the 'JCL – End User Dialogs Administration' panel, if you did not change the defaults in your profile or update the nickname table, your screen looks like this:

```
You may change the following JCL name for this request
WITHOUT affecting the defaults established in your profile.

JCL FILE NAME  ===> DVREDDXT
LOAD JCS NAME   ===> DVREDEJM or DVREDEJV

To display a list of existing JCL files,
enter a ? in either field.
```

Figure 80. JCL Review - End User Dialogs Aministration JCL fields for UIM

Otherwise, the UIM JCL that you specified in your profile or nickname table is displayed in the field. The JCS always defaults to DVREDEJM in MVS and DVREDEJV in VM.

2. Verify that DVREDDXT, or whichever UIM JCL file name you specified in your profile or in the nickname table, is the JCL file you want to use to send the request to the UIM. You should have edited this file using the JCL/JCS option from the Administrative Dialogs main menu.

3. Verify that DVREDEJM (for an MIT in MVS) or DVREDEJV (for an MIT in VM), is the JCS file you want to use to invoke the MIT Utility. You should have edited this file using the JCL/JCS option from the Administrative Dialogs main menu.

4. Press ENTER, after selecting the appropriate JCL and JCS files. You are now in an ISPF editing session.

An ISPF edit session is started on the MIT job stream. DataRefresher dialogs takes the information from the previous panels and internally creates a GETDEF command. The GETDEF command is imbedded in the JCL. For an explanation of the GETDEF command, see the DataRefresher Command Reference.

5. Edit the job stream, if necessary. The output of the job is the source for updating the master index table (MIT). The *E0 is within the allocation for DD name EUEIDS. There should be no statements other than *E0 in the input stream.
6. Press END (F3) when you have finished editing your UIM job. The ‘Send Confirmation’ panel is displayed. Fill in the appropriate fields and press ENTER. For information about the fields on the panel, press HELP (F1).

When your MIT resides in VM and you want to run the MIT utility on your own user space, you must edit the job returned from the UIM:

a. Remove the JOB card.
b. Remove the SET card.
c. Add a &TRACE ERR statement to convert this EXEC to an EXEC language that runs in your user space.
d. Create a file named EUEIDS FILE A on your A-disk.
e. Delete the FILEDEF INMOVE, FILEDEF OUTMOVE, and the MOVEFILE statements.
f. Move the data inserted by the UIM (at the *E0) into EUEIDS FILE.
g. Run the EXEC to invoke the MIT utility.

The job is sent to the UIM. The UIM generates an output job from the JCS and replaces the *E0 with the table description data.

If you have requested information once for each FDTLIB data you need to access, go to:

- “Editing a REM job to build or update the master index table (MIT),” if you need to include table descriptions for REM extract requests in your MIT.
- “Providing access to the master index table (MIT)” on page 148, if the above does not apply.

Editing a REM job to build or update the master index table (MIT)

If an end user needs to extract data from a DB2 or SQL/DS database, you must run a REM job to request the appropriate existing table descriptions and update the MIT for the end user:

1. On the ‘JCL – End User Dialogs Administration’ panel, if you did not change the defaults in your profile or update the nickname table, your screen looks like this:

```
You may change the following JCL name for this request
WITHOUT affecting the defaults established in your profile.

JCL FILE NAME  ==> DVREDREM or DVREDREV
LOAD JCS NAME  ==> DVREDEJM or DVREDEJV

To display a list of existing JCL files,
enter a 7 in either field.
```

Figure 81. JCL Review - End User Dialogs Administration JCL fields for REM

Otherwise, the default value of the JCL FILE NAME field is the name of the REM JCL that you specified in your profile or nickname table. The REM JCS always defaults to DVREDEJM or DVREDEJV.

2. Verify that DVREDREM (for DB2 data), DVREDREV (for SQL/DS data) or whichever REM JCL file you specified in your profile or in the nickname table, is the JCL file you want to use to send the request to the REM. You should have edited this file using the JCL/JCS option from the Administrative Dialogs
main menu, as described in Chapter 7, "Maintaining your JCL and JCS files" on page 65. The REM must be located on the same system as the data you want to extract.

3. Verify that DVREDEJM (for an MIT in MVS), DVREDEJV (for an MIT in VM), or whichever MIT JCS file you created is the JCS file you want to use to route the output back to the appropriate MIT. You should have edited this file using the JCL/JCS option from the Administrative Dialogs main menu, as described in Chapter 7, "Maintaining your JCL and JCS files" on page 65.

4. When you have selected the appropriate JCL and JCS, press ENTER.

You are now in an ISPF editing session. Your screen should look similar to the one shown below:

```
EDIT ------------------------------------- COLUMNS 001 072
COMMAND ===> SCROLL ===> PAGE
****** *********** TOP OF DATA ***********
000001 /*
000002 /*
000003 /*
000004 /*
000005 /* (MIT job stream goes here)
000006 /*
000007 /*
000008 /*
000009 /*
000010 /*
```

Figure 82. ISPF edit session panel

DataRefresher dialogs take the information from the panels and create a SUBMIT command, with an EXTRACT statement and a SELECT statement. This command is imbedded in the JCL. For an explanation of the SUBMIT/EXTRACT command and the SELECT statement, see the DataRefresher Command Reference.

5. Edit the job stream, if necessary. There should be no statements other than *E0 in the JCS input stream.

6. Press END (F3), when you have finished editing your REM job. The 'Send Confirmation' panel is displayed. Fill in the appropriate fields and press ENTER. For information about the fields on the panel, press HELP (F1).

When your MIT resides in VM and you want to run the MIT utility on your own user space, you must edit the job returned from the REM:

a. Remove the JOB statement.

b. Remove the SET statement.

c. Add a &TRACE ERR statement to convert this EXEC to an EXEC language that runs in your user space.

d. Create a file named EUEIDS FILE A on your A disk.

e. Delete the FILEDEF INMOVE, FILEDEF OUTMOVE, and the MOVEFILE statements.

f. Move the data inserted by the REM (at the *E0) into EUEIDS FILE.

g. Run the EXEC to invoke the MIT utility.
The REM generates an output job from the JCS contained in the data set with DD name EUEIJC5, and replaces the *E0 with the table description data. This becomes an inline data set that is the source for updating the MIT.

If you have requested information once for each relational system you need to access, continue to "Providing access to the master index table (MIT)."

**Maintaining the master index table (MIT)**

If you need to delete entries from the MIT, repeat the steps to request description data again. If you specify an asterisk (+) on the panel to request the names of the tables and associated views, all tables and associated views are deleted from the MIT for that source system; only the current data returned is stored in the MIT. As a result, the MIT is consistent with your DBMS catalog and FDTLIB.

You must have description data returned for at least one table with an associated view so that the MIT program has something to process.

You can run the MIT update program while your users are using the End User Dialogs. The updated information is not available to your users until they exit DataRefresher dialogs and ISPF, then restart a DataRefresher dialogs session.

**Providing access to the master index table (MIT)**

Your users are ready to be enrolled so they can start using the End User Dialogs by accessing the MIT just updated or created.

Your users run a CLIST or EXEC when they log on, as discussed in Chapter 6, "Starting the Administrative Dialogs" on page 63. Among other things, this CLIST/EXEC provides the user access to the MIT in the ISPTLIB DD name.

The CLIST provides the user access to the MIT by allocating DVRTADME in the concatenation on the ISPTLIB DD statement. The EXEC provides the user with access to the MIT by a library definition for DVRTADME in the concatenation to the ISPTLIB DD name. An example of what the EXEC and CLIST might look like is shown in the *DataRefresher Administration Guide*.

Without the proper access to the MIT, users cannot to build any DataRefresher End User Dialogs extract requests.

You are ready to either:

- Start using the End User Dialogs yourself
- Enroll other users as explained in:
  - "Enrolling MVS DataRefresher dialogs users" on page 149
  - "Enrolling VM DataRefresher dialogs users" on page 151
Enrolling users

The following sections describe how to enroll DataRefresher dialogs users on the MVS and VM systems.

Enrolling MVS DataRefresher dialogs users

To provide one or more new users with a complete MVS DataRefresher dialogs environment:

1. Invoke the DVRELNRL CLIST, as described in the DataRefresher Administration Guide. When prompted by the CLIST, enter the user ID of the user or users you want to enroll.

   DVRELNRL builds the following data sets for the users being enrolled:
   
   - &userid.DVR110.DVRTLIBE
   - &userid.DVR110.DVRDPROF
   - &userid.DVR110.DVRIMEXE
   - &userid.DVR110.DVJRJEDIE

   Where userid is the user's user ID.

2. In Chapter 15, "Providing access to dialog objects" on page 135, you chose how your users access DataRefresher dialogs objects according to your installation. If you chose to use object sharing, or if you edited the default dialogs models without changing the names, go to step 8, skipping the intermediate steps.

   If you chose to build a new file from the dialogs models, go to the next step. You must use the Administrative Dialogs to import the edited and renamed models. (You already exported the models in Chapter 15, "Providing access to dialog objects" on page 135.)

   If your user is allowed to run Administrative Dialogs, you may have the user perform the remaining steps in this section.

3. Log on to the user's ID, or have the user do this if the user is allowed to run Administrative Dialogs. You must be able to access the administrator's disk in read mode to import dialogs objects from the Import/Export library.

4. Initiate a DataRefresher dialogs session by running DVRELLLGN CLIST (explained in Chapter 6, "Starting the Administrative Dialogs" on page 63).

5. Select option 3, JCL/JCS, on the Administrative Dialogs main menu:

6. Select option 2, Import or Export ..., on the 'Build and Maintain JCL Files' panel.

```
BUILD AND MAINTAIN JCL FILES

SELECT OPTION ---> 2

1.   EDIT, BUILD, or ERASE a JCL/JCS file.
2.   IMPORT or EXPORT a JCL/JCS file.
```

Figure 83. Build and Maintain JCL files panel
7. Select option 1, Import a DataRefresher dialogs object, on the ‘Select Import/Export Function’ panel to import the objects that your end users need from the administrator’s Import/Export library.

```
SELECT IMPORT/EXPORT FUNCTION
SELECT OPTION ===> 1
  1 IMPORT a DataRefresher Dialogs object
  2 EXPORT a DataRefresher Dialogs object
  3 ERASE object(s) from Import/Export library

Enter the name of the dataset that you want to import from or export to. This must be an existing dataset. Enclose the name in single quotes if you do not want a prefix inserted.

DATASET NAME: DVR110.DVRIMEXE
```

Figure 84. Select Import/Export Function panel

The default Import/Export library name is DVR110.DVRIMEXE.

8. Enter S next to the file you want to import on the list of DataRefresher dialogs objects in the Import/Export library.

```
COMMAND ===> IMPORT/EXPORT - EXTERNAL MEMBER LIST
SCROLL ===> HALF

Type the appropriate letter in the SEL column on ONE line only and press ENTER.

  s = select member name

Press: HELP key for information ENTER to continue END key to return

SEL MEMBER NAME
```

Figure 85. Import/Export - External Member List

**Note:** You may only import one file at a time.

9. Specify whether the file is JCL or JCS by selecting the appropriate option on the following panel:

```
SELECT OPTION ===> IMPORT - JCL/JCS

Select the option that describes the JCL/JCS FILE you are importing.

  1 JCL
  2 JCS
```

Figure 86. Import - JCL/JCS
10. Specify a name and description of the file you are importing on the following panel and press ENTER:

```
SELECT OPTION ----> IDENTIFY JCL/JCS NAME

Create and enter a NAME for the JCL/JCS you are importing.
Enter a ? in NAME field to display a list of existing objects.
The DESCRIPTION field is optional but highly recommended.

NAME ---->
DESCRIPTION ---->
```

**Figure 87. Identify JCL/JCS Name**

You return to the Select Import/Export Function panel and a message telling you that the import dialog has completed successfully is displayed.

Repeat the above steps to import objects until you have imported all the necessary objects for the end user. In most cases the profile (&userid.DVR110.DVRDPROF) is sufficient for the user to build and send requests. However, some users have special needs, like sending a request to a different system. If you have to change a user's profile, you must do it from the user's user ID as discussed in Chapter 14, “Changing the default profile” on page 127.

You are now ready to use the End User Dialogs. If you have not already initiated a DataRefresher dialogs session, see Chapter 6, “Starting the Administrative Dialogs” on page 63.

**Enrolling VM DataRefresher dialogs users**

To provide a new user with a complete VM DataRefresher dialogs environment:

1. Invoke the DVREXLIB EXEC, as explained in the *DataRefresher Administration Guide*. When prompted by the EXEC, enter the user ID of the user(s) you want to enroll.

   DVREXLIB sends the following files to the newly enrolled user:
   - DVRJEDIE MACLIB
   - DVRIMEXE MACLIB
   - *userid* MACLIB (profile)

2. In Chapter 15, “Providing access to dialog objects” on page 135, you chose how your users access DataRefresher dialogs objects according to your installation. If you chose to use object sharing, or if you edited the default dialogs models without changing the names, go to step 9, skipping the intermediate steps.

   If you chose to build a new file from the dialogs models, go to the next step. You must use the Administrative Dialogs to Import the edited and renamed models. (You already exported the models in Chapter 15, “Providing access to dialog objects” on page 135).

   If your user is allowed to run Administrative Dialogs, you can have the user perform the remaining steps in this section.

3. Log on to the user's ID. You must be able to access the administrator's disk in read mode to import dialogs objects from the Import/Export library.
4. Receive DVRJEDIE MACLIB, DVRIMEXE MACLIB, and userid MACLIB from the reader.

5. Initiate a DataRefresher dialogs session by running DVREXUSR EXEC (explained in Chapter 6, "Starting the Administrative Dialogs" on page 63).

6. Select option 3, JCL/JCS, from the Administrative Dialogs main menu:

7. Select option 2, Import or Export a file on the 'Build and Maintain JCL Files' panel:

```
SELECT OPTION ===> 2
   BUILD AND MAINTAIN JCL FILES
      1   EDIT , BUILD , or ERASE a JCL/JCS file.
      2   IMPORT or EXPORT a JCL/JCS file.
```

**Figure 88. Build and Maintain JCL files panel**

8. Select option 1, Import a DataRefresher Dialogs object, on the 'Select Import/Export Function' panel to import the objects your end users need from the administrator's Import/Export library.

```
SELECT IMPORT/EXPORT FUNCTION
SELECT OPTION ===> 1
   1   IMPORT a DataRefresher Dialogs object
   2   EXPORT a DataRefresher Dialogs object
   3   ERASE object(s) from Import/Export library

Enter the name of the library that you want to import from or export to. This must be an existing library, and file type must be MACLIB.

LIBRARY file name : DVRIMEXE
file mode : A
```

**Figure 89. Select Import/Export Function**

The 'Import/Export – External Member List' panel is displayed. This panel contains a list of DataRefresher dialogs objects in the Import/Export library.

9. Type an S next to the file you want to import, and press ENTER.

**Note:** You may only import one file at a time.

```
COMMAND ===> IMPORT/EXPORT – EXTERNAL MEMBER LIST
ROW 1 OF 7
SCROLL ===> HALF
Type the appropriate letter in the SEL column on ONE line only and press ENTER.
   s = select member name

Press: HELP key for information  ENTER to continue  END key to return
SEL MEMBER NAME
```

**Figure 90. Import/Export - External Member List**
10. Specify whether the file is JCL or JCS by selecting option 1 for JCL or option 2 for JCS on the 'Import - JCL/JCS' panel.

![Image](import-jcl-jcs.png)

**Figure 91. Import - JCL/JCS**

11. Specify a name and description of the file you are importing and press ENTER.

![Image](identify-jcl-jcs-name.png)

**Figure 92. Identify JCL/JCS Name**

You return to the 'Select Import/Export Function' panel and a message telling you that the import dialog has completed successfully is displayed.

Repeat the above steps, until you have imported all the necessary objects for the user or users. In most cases the profile (userid MACLIB, member DVRDPROF) is sufficient for the user to build and send requests. However, some users have special needs, like sending a request to a different system. If you need to change a user's profile, you need to do it from the user's user ID as discussed in Chapter 14, "Changing the default profile" on page 127.

You may need to provide the DVREXSND EXEC to certain end users. If your user also wants to use the Administrative Dialogs, you should probably provide the user with an edited copy of DVREXSND EXEC (see the DataRefresher Administration Guide).

If you did not import the models and you are not migrating, instruct your users to receive the following files from their readers:

- DVRJEDIE MACLIB
- DVRIMEXE MACLIB
- userid MACLIB

You, or the user, can now initiate a DataRefresher dialogs session by running DVREXUSR EXEC, as explained in Chapter 6, "Starting the Administrative Dialogs" on page 63.
Part 3. Using the End User Dialogs
Chapter 17. What are End User Dialogs?

End User Dialogs make data extracts easier by letting you control DataRefresher through dialog panels rather than through coding individual jobs and commands. The panels help you fill in the proper information on the screen and build an extract request.

Before you can use End User Dialogs, your administrator must set them up for you. For information about setting up the End User Dialogs, see Chapter 16, “Setting up End User Dialogs” on page 139.

End User Dialogs let you specify the characteristics of an extract request. The panels ask you detailed questions about the data you want to extract.

Beginners or infrequent users may find End User Dialogs simpler to learn than writing an extract. Such users could be programmers or project leaders who need sample data, or other professionals who need data for a report.

In fact, all DataRefresher users not familiar with DataRefresher commands might find End User Dialogs simpler to use, because the dialog panels collect information used to generate DataRefresher commands that build an extract request for the user.

At the minimum, you should know enough about Interactive System Productivity Facility (ISPF) to start an End User Dialogs session and to check the job output for a successful run (the extract built through the dialogs is a batch job).

Working with End User Dialogs

These are the steps you take to extract data using the End User Dialogs:

1. Plan the extract
2. Build the extract
3. Provide database access information
4. Save the extract
5. Send the extract

The following chapters explain these steps and provide some examples of how to use End User Dialogs to extract data from a source and send the data to a target database.

Navigating through the End User Dialogs

End User Dialogs consist of a series of panels that let you build and send extract requests.

When you first invoke End User Dialogs, as described in Chapter 18, “Starting the End User Dialogs” on page 163, the DataRefresher copyright panel is displayed. Press ENTER to display the main menu panel, which is shown in Figure 93 on page 158.

This menu provides you with access to the End User Dialogs options. When you select an option, or use a function key to access an option, the panel for that option is displayed.
DATAREFRESHER END USER DIALOGS

Select ONE of the following options,
or use the PF Keys:

1  TABLES  - Display table and DXTVIEW names available for extract
2  COLUMNS - Display column names of tables or DXTVIEWs selected
3  CONDITIONS - Specify conditions on the columns
4  JOIN - Specify join condition if more than one table or DXTVIEW selected
5  TARGET - Specify the target for loading extracted data
6  DB ACCESS - Specify database access information
7  PROFILE REVIEW - Change Profile settings

Available Commands: Send, Save, Display, Status, Cancel, Reset, Check, Erase

OPTION --->
F1=HELP    F2=SEND    F3=END    F4=TABLES    F5=COLUMNS    F6=CONDITION
F7=BACKWARD  F8=FORWARD  F9=JOIN    F10=TARGET    F11=EXT LIST  F12=ACCESS

Figure 93. DataRefresher End User Dialogs main menu

The following options are available from the End User Dialogs main menu:

1 Tables
Displays a list of available tables and DXTVIEWs to use as source data for your request. Tables and DXTVIEWs are different ways to represent source data. Tables describe relational data; DXTVIEWs represent non-relational data.

2 Columns
Displays the column names for the table(s) or DXTVIEW(s) you selected with the Tables options.

Note: The order in which the columns are displayed is determined by the database manager. Therefore, when you build a request, what is displayed on your screen may appear different from what is displayed on the screens in this book.

3 Conditions
Displays the column names of the selected tables or DXTVIEWs. These tables or views were selected with the Tables option. You can specify conditional selection criteria on one or more columns.

In DataRefresher a condition is a statement about the relationship between two or more strings of characters or numbers that must be satisfied before extraction from a row or record occurs.

4 Join
Displays the tables and DXTVIEWs and their associated columns. You can now specify a join condition. This panel is used when you choose multiple tables or DXTVIEWs for the extract.

5 Target
Lets you specify the target system, name a target table, and name and order columns for your extract request.
6 DB Access
Lets you provide information necessary to access source databases.

7 Profile Review
Lets you change the processing and output options that DataRefresher uses for the extract request when the request is saved or sent.

Figure 94 shows how you can navigate through the options to complete an extract request. If you are unsure of your next step, that is you do not know which option to continue with, enter the CHECK command. This command is used to ensure that the information necessary for the successful execution of an extract request has been entered in this session.

![Diagram showing End User Dialogs option structure]

**End User Dialogs function keys**
The End User Dialogs provide a series of function keys that can be used to access the options available on the End User Dialogs main menu. The following is a list of these PF keys.

**ACCESS (F12)**
Displays the 'Database Access' panel, to let you enter your database access information for the source data.

**Note:** This PF key is not available when you can scroll left and right on a panel. For example, with the 'Name Target for Extract Output' panel.

**BACKWARD (F7)**
Scrolls backwards in a panel containing more rows than can be displayed on a screen.

**COLUMNS (F5)**
Displays the 'Select Columns for Extract' panel, so that you can select the columns to be used from a DXTVIEW or Table.
CONDATION (F6)
Displays the 'Specify Conditions' panel, so that you can specify the conditions that must be met if the data is to be extracted.

END (F3)
Ends the display of the panel while saving any changes to the panel, and returns to the previous panel or the End User Dialogs main menu, depending on the panel being displayed.

EXT LIST (F11)
Displays the 'Display Extract Request' panel, so that you can select an extract request.

Note: This PF key is not available when you can scroll left and right on a panel. For example, with the 'Name Target for Extract Output' panel.

FORWARD (F8)
Scrolls forward in a panel containing more rows than can be displayed on a screen.

HELP (F1)
Displays either panel help or message help.

JOIN (F9)
Displays the 'Specify Joins' panel, to select the columns which are to be joined from multiple tables.

LEFT (F11)
Scrolls left in a panel containing more fields than can be displayed on a screen.

RIGHT (F12)
Scrolls right in a panel containing more fields than can be displayed on a screen.

SEND (F2)
Displays the 'Send Confirmation' panel, so that you can send the currently active extract request for processing.

TABLES (F4)
Displays the 'Select Tables/DXTVIEWS for Extract' panel, containing a list of the Tables and DXTVIEWS that can used in an extract request.

TARGET (F10)
Displays the 'Name Target for Extract Output' panel, for identifying the target table that receives the extracted data.

End User Dialogs commands
You can enter DataRefresher End User Dialogs commands on the command line. These commands are listed on the End User Dialogs main menu above the function key descriptions, as shown in Figure 93 on page 158. The commands are:

- CANCEL
- CHECK
- DISPLAY
- ERASE
- RESET
- SAVE
- SEND
• STATUS

For detailed descriptions of these commands, see the DataRefresher Command Reference.

ISPF commands
You can type ISPF commands on the End User Dialogs command line. For example, if the ISPF command END is not available from a function key, you must type the command and then press ENTER. ISPF commands are described in Interactive System Productivity Facility Dialog Management Guide and Reference.

Getting help

While using the End User Dialogs, you can get help from messages or panel help.

Panel help
You can get to panel help either by pressing HELP (F1) with no error message displayed, or by pressing ENTER on the last message help panel. When you first enter panel help, a menu is displayed with a table of contents. To get help on any one of the categories, type its number on the command line and press ENTER.

Message help
If a message is displayed during an End User Dialogs session, you can get more information about the message by pressing HELP (F1), or by typing HELP on the command line and pressing ENTER. An error message help panel is displayed. You can then press END (F3) to return to the dialog panel, or press ENTER to see panel help.

Using nicknames

A nickname is an easy way to refer to a particular location and to the JCL or JCS that routes a request or output to that location. During an End User Dialogs session, you may use nicknames to specify the source or target system of the data. To use nicknames, your administrator must set up and build entries in a nickname table

For more information about setting up a nickname table, see “Building or updating the nickname table” on page 140.

If you do not use nicknames, you must complete the 'Profile Review' panel to supply the file names that route a request or output, or the administrator must specify the information in your profile.

You may want to use nicknames because they:

• Are easier to remember than system names.

• Eliminate the need to change information in the 'Profile Review' panel when you want to process extract requests that have different source or target systems.

• Simplify specifying subsystems on a node.
Setting up End User Dialogs

Before an End User Dialogs extract request can run successfully, certain administrative tasks (such as writing instructions to link to the proper source database or creating a target table) must be completed by a DataRefresher administrator.

Additional tasks must be done to move data from source to target. The following files should be built by your DataRefresher administrator:

- Job control language (JCL) files that route the extract request definition to the source system and start the extract program.
- Job control language (JCL) files that start the extract program.
- Job control statement (JCS) files that control the extracted data and move it to the correct target in the proper format.

When you extract non-relational data, you also need to write descriptions of the data you want to extract. You do not need to know how to build JCL/JCS files or write data descriptions to use the End User Dialogs, but you do depend on a DataRefresher administrator to build the files that support your extract request. Otherwise, your extract request does not have the supporting code to successfully extract your data.

Other administrative tasks include controlling:

- Who can use the End User Dialogs
- What source data can be extracted
- What JCL or JCS files the end user can access

This is done by building the master index table and controlling the user profile. For more information, see "Editing a UIM job to build or update the master index table (MIT)" on page 145 or Chapter 14, "Changing the default profile" on page 127.

When the above administrative tasks are done, even with little training you can run your own extracts to extract data from a source to a target location. You must understand the options on the dialog panels, but you do not need to understand every DataRefresher command. Without the End User Dialogs, you would need to know how to write an extract request without prompting.
Chapter 18. Starting the End User Dialogs

This chapter describes how to start a session with the End User Dialogs by invoking the DVRELLGN CLIST in an MVS environment, or DVREXUSR EXEC in a VM environment.

A description of how to set up the DVRELLGN CLIST and the DVREXUSR EXEC is provided in the DataRefresher Administration Guide.

Starting an MVS dialog session
Use the following steps to invoke the DVRELLGN CLIST, which starts an MVS DataRefresher dialogs session:

1. Run the ISPF logon procedure. This procedure sets up the ISPF environment and invokes the ISPF main menu.

2. Select the MVS DataRefresher dialogs option from the ISPF menu.

3. A message may be displayed asking whether you want to use the DataRefresher sample data. This message is only displayed if the DataRefresher sample data is available in your environment.
   
   Type Y and press ENTER to use the sample data, or press ENTER if you do not want to use the sample data.

4. A message may be displayed asking whether you want this dialog session to be an object sharing session. This message is only displayed if object sharing is available in your environment.
   
   Type Y to use object sharing, or N if you do not want to use object sharing, and press ENTER.

When the CLIST is complete, the DataRefresher copyright panel is displayed. Press ENTER, to display the End User Dialogs main menu.

Starting a VM dialog session
Use the following steps to invoke the DVREXUSR EXEC, which starts a VM DataRefresher dialogs session:

1. Run the ISPF invocation EXEC (provided by the administrator).

   This EXEC should set up the ISPF environment and invoke the ISPF main menu; it assumes that the DataRefresher production disk has already been accessed.

2. Select the End User Dialogs option from the ISPF menu.

3. A message may be displayed asking whether you want to use the DataRefresher sample data. This message is only displayed if the DataRefresher sample data is available in your environment.

   Type Y and press ENTER to use the sample data, or press ENTER if you do not want to use the sample data.

4. A message may be displayed asking whether you want this dialog session to be an object sharing session. This message is only displayed if object sharing is available in your environment.
Type Y to use object sharing, or N if you do not want to use object sharing, and press ENTER.

When the EXEC is complete, the DataRefresher copyright panel is displayed. Press ENTER, to display the End User Dialogs Main Menu.
Chapter 19. Extracting data from a non-relational source

This chapter provides an example of how to extract data from a non-relational source, and place the data in a relational source. The samples shown in this example are based on an IMS data source, with the data being extracted from an IMS database called IMSPROJECT, and a DB2 target. However, the actions you take to complete this example extract are the same for any non-relational extract.

To extract the data from the data source and place it in the target database, the following DataRefresher administrative tasks must have been completed:

- A description of the source data must be stored in the FDTLIB
- The master index table (MIT) must be updated to include the data source
- The JCL and JCS required to run the DEM and load the extracted data into the target must have been created

When the administrator has completed these tasks, you must perform the following tasks to create the extract:

1. Build the extract:
   - Start the End User Dialogs session
   - Select the DXTVIEW that contains the source data you want
   - Select the columns you want from the DataRefresher view
   - Specify the conditions the source data must meet
   - Identify the target table for your extracted data

2. Provide database access information

3. Save the extract

4. Send the extract to be processed

Note: This example assumes the function keys F1 through F12 are set for DataRefresher. Your installation may use function keys F13 through F24 for DataRefresher; if so press F13, instead of F1, F14 for F2, and so on.

Task 1. Building the extract request

You build an Extract Request using End User Dialogs by filling in the End User Dialogs panels, identifying the data you want to extract and where you want to move it.

Step 1. Starting an End User Dialogs session

After logging on to your system, start ISPF so that the ISPF/PDF Primary Option Menu is displayed, and start the End User Dialogs as described in Chapter 18, “Starting the End User Dialogs” on page 163. The End User Dialogs main menu is displayed when you start the dialogs.
Figure 95. DataRefresher End User Dialogs main menu

This panel provides you with the options required to build an extract request. In the example in this chapter, you use the Tables, Columns, Conditions, Target, and DB Access options.

Step 2. Selecting a DXTVIEW

To select the view you want to use in your extract:

1. Select the Table option from the End User Dialogs main menu, by pressing F4 (Tables) or typing 1 in the option line at the bottom of the menu and pressing ENTER.

2. The 'Select Tables/DXTVIEWs for Extract' panel is displayed. The screen displayed in your installation should be similar to the following screen.
The panel shows the sample tables and DXTVIEWs you can extract using End User Dialogs.

- Data stored in relational databases is represented as tables which are listed by Table Name.
- Data stored in non-relational data sources (IMS/VS DL/I databases; VSAM or physical sequential files), or accessed through generic data interface (GDI) exits (like remote data sources), are listed as DXTVIEWs.

Each DXTVIEW and table is listed on a separate line, with seven descriptive fields for each line. Note that some of the fields are not shown, you need to page to the right, by pressing F12 (Right), to display the rest of the fields. To return to the first field, press F11 (Left). Help is available for each field, by pressing F1 (Help).

You can use F8 (Forward) and F7 (Backward), to page through the list of tables and DXTVIEWs.

3. To see the Description column, press F12 (Right).

4. Press PF11 (Left), to return to the Name column.

5. Type S in the SELECT column next to the description for IMSPROJECT.

Now you can select the columns from the selected view, go to “Step 3. Selecting columns from the DXTVIEW” on page 168.

Note: Your selections are saved while you remain in this End User Dialogs session. Except for the 'Database Access' panel, pressing ENTER does not affect your input, nor does it bring up the next dialog screen.
Step 3. Selecting columns from the DXTVIEW

Once you have selected your DXTVIEW, or table, you can select the columns you want to use from the DXTVIEW or table.

1. Press PF5 (Columns), to display the 'Select Columns for Extract' panel, as shown below.

```
SELECT COLUMNS FOR EXTRACT
Do you want to select ALL columns for extract, Yes or No? ===> N
If not, enter an s in the SELECT column of the desired column name.
To view additional column information, scroll RIGHT.

SELECT COLUMN   DXTVIEW
  S    NAME
  =>    DEPT    IMSPROJECT
  =>    DEPTNAME    IMSPROJECT
  =>    MANAGER    IMSPROJECT
  =>    DIVISION    IMSPROJECT
  =>    LOCATION    IMSPROJECT
  =>    PROJNUM    IMSPROJECT
  =>    PROJNUM    IMSPROJECT
  =>    STARTD    IMSPROJECT
  =>    ENDD    IMSPROJECT
  =>    TIMESTAMP    IMSPROJECT

COMMAND ===>   SCROLL ===> PAGE
 F1=HELP  F2=SEND  F3=END  F4=TABLES  F5=COLUMNS  F6=CONDITION
 F7=BACKWARD  F8=FORWARD  F9=JOIN  F10=TARGET  F11=LEFT  F12=RIGHT
```

Figure 97. Select Columns for Extract panel

Each line on this panel corresponds to a column in the selected DXTVIEW, IMSPROJECT. There are four descriptive fields for each column. To display the meanings of the fields, press F1 (Help).

You can use F8 (Forward) and F7 (Backward), to page through the list of columns.

2. Press F12 (Right) to look at the Description field.

3. If you want to select all the columns, enter Y in the first line on the panel.

To select several columns, leave the default value N in the first line, and type an S in the Select field next to each column you want to select.

**Note:** Ensure that you select the DEPTNAME column, to continue with this example.

Step 4. Specifying the source data conditions

Once you have selected your columns from the DXTVIEW or table, you can specify any conditions that must be met before data is extracted.

1. Press F6 (Condition) to display the 'Specify Conditions' panel.
Figure 98. Specify Conditions panel

The panel shows an entry for each column selected in the previous step. You can control the information accessed from these columns by specifying conditions that must be met before the data is extracted. For information about each of the fields on the panel, press F1 (Help). You can use F8 (Forward) and F7 (Backward), to page through the list of columns.

In this example, you only want to display information on projects controlled by the head office. DataRefresher should only extract data entries with Head Office, in the DEPTNAME column.

2. Ensure that the column DEPTNAME is displayed on the panel.

3. Type = Head Office in the DEPTNAME Condition field.

Figure 99. Specify Conditions DEPTNAME field

4. Press ENTER.

DataRefresher automatically creates a copy of the row below the origin, letting you enter multiple conditions for the same column. DataRefresher connects multiple conditions with "AND".

When you finish specifying conditions you have to identify the target table.
Step 5. Identifying the target table

Once you have specified your conditions for the columns in the DXTVIEW or table, you have to identify the target table, that will receive the extracted data.

1. Press F10 (Target) to display the 'Specify Conditions' panel,

2. The 'Name Target for Extract Output' panel is displayed. The following figure shows an example of this panel.

![Figure 100. Name Target for Extract Output left panel](image)

For the meaning of the fields on this panel, press F1 (Help).

3. Type the following information in the top of this panel.

   - **Target Nickname**: See "Using nicknames" on page 161 for information on using nicknames.
   - **Target Table Name**: Only make an entry for this field if the target system is a relational system or an IXF data set or file.
   - **Table Qualifier**: The prefix of the target table.
   - **Target Table Option**: Only make an entry for this field if the target system is a relational system.

If you do not know the DataRefresher nickname you should specify, you can type a question mark (?) in the Target Nickname field. The 'List of Nicknames' panel is displayed. This panel lists the contents of the nickname table.

To select a nickname from the list:
   a. Type an S in the SEL column next to the nickname.
   b. Press F3 (End) to return to 'Name Target for Extract Output' panel.

The Target Nickname field contains the selected nickname.

4. Use the Target Column Name field to identify the target column where each source column's data must be sent.

You only use the field, when the entry for the field is not the same as the entry for the Input Column Name field.
5. Specify whether the target column accepts a null value. The DB2 load utility, which loads the data into the target table, requires this information.

If a target column does not accept a null value replace the Y in the Accepts Null field with an N.

6. Press F12 (Right) to see the right side of the panel. Your screen should be the following figure:

```
NAME TARGET FOR EXTRACT OUTPUT
Row 1 of 7
Target Nickname ===> Specify nickname of target system
Target Table Name ===> Specify for DB2, SQL/DS & IXF
Table Qualifier ===> Prefix for target table name
Specify when Target is SQL/DS or DB2:
Target Table Option ===> C / R / blank

Change the TARGET COLUMN NAMEs where appropriate.

INPUT COLUMN ORDER SEQUENCE OUTPUT ORDER
NAME (1 - 8) (ASC/DESC)
LOCATION => 1 => ASC
STARTD => =>
ENDD => =>

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

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

COMMAND ===> SCROLL ===> PAGE
F1=HELP F2=SEND F3=END F4=TABLES F5=COLUMNS F6=CONDITION
F7=BACKWARD F8=FORWARD F9=JOIN F10=TARGET F11=LEFT F12=RIGHT

Figure 101. Name Target for Extract Output right panel
```

7. Specify the organization of the of the output columns, by entering an Order Sequence and Output Order (Ascending or Descending).

In the example panel above, the output is sorted by location in ascending order.

8. Press F3 (End) to return to the main menu.
Task 2. Providing database access information

To provide DataRefresher with access to the data, press F12 (Access) when the End User Dialogs main menu is displayed. The 'Database Access' panel is displayed.

<table>
<thead>
<tr>
<th>DATABASE ACCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source information:</td>
</tr>
<tr>
<td>Location === NYC1</td>
</tr>
<tr>
<td>User ID === YOURID</td>
</tr>
<tr>
<td>Password ===</td>
</tr>
<tr>
<td>VM Execution ID ===</td>
</tr>
<tr>
<td>Target information when source is SQL/DS:</td>
</tr>
<tr>
<td>User ID ===</td>
</tr>
<tr>
<td>Network ID ===</td>
</tr>
<tr>
<td>Target Node ID === CHIMWS1</td>
</tr>
<tr>
<td>Press ENTER to save the required information.</td>
</tr>
</tbody>
</table>

COMMAND ===>
F1=HELP F2=SEND F3=END F4=TABLES F5=COLUMNS F6=CONDITION
F7=BACKWARD F8=FORWARD F9=JOIN F10=TARGET F11=EXT LIST F12=ACCESS

Figure 102. Database Access panel

When the panel is displayed it contains entries for the Location, User ID, and Target Node ID fields.

Note: The 'Database Access' panel is the only panel that saves your entries when you press ENTER. For the meaning of the fields on this panel, press F1 (Help).

When you extract data from an SQL/DS table, you must provide additional information for routing the output job. This information includes the user ID and the target node ID to which your extract output is routed. When you extract data from other sources, your administrator provides this information in the JCS.

You use the panel to create an entry in DataRefresher for each source node from which you want to extract data. DataRefresher saves database access information separately from the Extract Requests, so you do not have to repeat the information for this particular source node.

To create the entry:

1. Replace the displayed User ID, with your User ID for the source node, if it is different.
2. Type your password for the User ID. What you type is not displayed.
3. Press ENTER. A message is displayed above the command line, informing you that the access table has been updated.

If you select any table from the same node as your source table, when you are building another request, DataRefresher displays this access entry on the 'Database Access' panel.
You can change the entries by changing the entries on this panel. Your entries replace the information previously stored in your Database Access table.

Task 3. Saving the extract request

When you have completed the information required for the previous tasks on the Tables, Columns, Conditions, Target, and 'Database Access' panels, you can add your Extract Request to the Dialogs library. Once you have saved an extract, it can be used at any time in the future.

The name you use for saving the Extract Request must be unique, and no more than 18 characters long. You can check whether a name is unique by pressing F11 (Ext List) or entering the command DISPLAY on the command line before you save the request. When you do this, the 'Display Extract Requests' panel is displayed. This panel contains a list of the extract requests that already exist in your library.

If you are in an object sharing session, only names contained in the common object library appear on the panel. You can check through the list of saved Extract Requests, to ensure that the name you are entering is unique.

To save the Extract Request you have built, enter the following command in the command line:

```
SAVE AS extract (DESC = 'description')
```

Where:

- `extract` is the name of the extract.
- `description` is a brief description of the extract request. The description must be enclosed in single quotes, and can be up to 31 characters long.

This description is optional, but it might help you to identify the extract later.

After entering the SAVE command, you can press F11 (Ext List) or enter the DISPLAY command in the command line to display the 'Display Extract Request' panel, and verify that DataRefresher saved your request.

**Note:** When you display this panel you can add or change the description for an Extract Request, by entering a new description in the Description field.

If you specified object sharing when you logged on to the End User Dialogs, your saved request is placed in a shared library containing extract requests. You must have WRITE access to save Extract Requests to the shared library.

If you do not have WRITE access to the shared library, or if you are not accessing the shared library in this End User Dialogs session, your saved Extract Request now resides in your personal library of Extract Requests.
Task 4. Sending the extract request for processing

At this point, you have:

- Built an extract request
- Provided DataRefresher with the access information
- Saved the extract request

Also, your administrator has created the JCL and JCS you need to use for this extract request. These JCL and JCS files must be specified on your 'Profile Review' panel. The information specified on this panel is saved with your request. If you are unsure whether the JCL and JCS files specified in your profile are correct, check the 'Profile Review' panel and consult your DataRefresher administrator.

To send the request for processing:

1. Enter the SEND command followed by the name of the request. For example:
   
   SEND extractname

   **Note:** You can enter the command without specifying the name of a request. If you do this DataRefresher will send the currently active request for processing. You make a request active by creating it, or selecting it from the 'Display Extract Request' panel.

   The 'Send Confirmation' panel is displayed.

   ![Send Confirmation Panel](image)

   **Figure 103. Send Confirmation panel**

2. To confirm that the request is to be sent for processing, press ENTER.

The extract request is sent for processing, and you can:

- Start another End User Dialogs session:
  
  1. Press F4 (TABLES) to return to the 'Select Tables/DXTVIEWS for Extract' panel.
  
  2. Type RESET on the command line and press ENTER. The RESET command clears all selections from the previous request.
  
  3. Select the table or DXTVIEW for another dialog.

- Return to the ISPF/PDF Primary Option Menu to start an Administrative Dialogs session:
1. Press F3 (END) to return to the End User Dialogs main menu.
2. Press F3 (END) to return to the ISPF Primary Option menu.
3. Select the DataRefresher Administrative Dialogs option on the menu
   • Exit the DataRefresher dialogs:
     1. Press F3 (END) to return to the End User Dialogs main menu.
     2. Press F3 (END) to return to the ISPF Primary Option menu.
     3. Choose another option on the ISPF/PDF Primary Option Menu, or exit ISPF
        by pressing F3 (END).
Chapter 20. Extracting from DB2 to SQL/DS

This example shows you how to extract data from two DB2 tables and place the data into an SQL/DS table using End User Dialogs. This way you can create a new table in VM from information in two tables in MVS. The sample data extracted is called DB2STAFF and DB2ORG.

To extract the data from the DB2 sources and place it in the target SQL/DS database, the following DataRefresher administrator tasks must have been completed:

- The master index table for the data sources, DB2STAFF and DB2ORG, must be updated. For more information, see "Editing a UIM job to build or update the master index table (MIT)" on page 145.
- The JCL required to link to the correct DB2 library and to start the REM, and the JCS required to link to the correct SQL/DS target must have been created.

When the administrator has completed these tasks, you must perform the following tasks to create the extract:

1. Build the extract:
   - Start the End User Dialogs session.
   - Choose the source tables that contain the data you want.
   - Select the columns you want from the source tables, and the built-in functions needed to produce the columns you want.
   - Specify the conditions the source data must meet.
   - Specify where to join two rows from the source tables to form a row in the target table.
   - Identify the target table that is to receive the extracted data.

2. Provide the database access information

3. Save the extract

4. Send the extract

Note: This example assumes that function keys F1 through F12 are set for DataRefresher. Your installation may use function keys from F13 through F24 for DataRefresher; if so press F13 instead of F1, F14 for F2, and so on.

Task 1. Building the extract request

You build an extract request using End User Dialogs by filling in panels to identify what data you want to extract and where you want to move it.

Step 1. Starting the End User Dialogs session

After your information center or DataRefresher administrator shows you how to log on to your system and get to the 'ISPF/PDF Primary Option Menu', select the option to start the End User Dialogs session. This is described in Chapter 18, "Starting the End User Dialogs" on page 163.
When you start the End User Dialogs session, the IBM copyright panel appears. Press ENTER to display the End User Dialogs main menu panel:

**DATAREFRESHER END USER DIALOGS**

Select ONE of the following options, or use the function keys:

1. **TABLES** - Display table and DXTVIEW names available for extract
2. **COLUMNS** - Display column names of tables or DXTVIEWS selected
3. **CONDITIONS** - Specify conditions on the columns
4. **JOIN** - Specify join condition if more than one table or DXTVIEW selected
5. **TARGET** - Specify the target for loading extracted data
6. **DB ACCESS** - Specify database access information
7. **PROFILE REVIEW** - Change Profile settings

Available Commands: Send, Save, Display, Status, Cancel, Reset, Check, Erase

**OPTION ===>
F1=HELP  F2=SEND  F3=END  F4=TABLES  F5=COLUMNS  F6=CONDITION
F7=BACKWARD  F8=FORWARD  F9=JOIN  F10=TARGET  F11=EXT LIST  F12=ACCESS

Figure 104. DataRefresher End User Dialogs main menu

Use the options listed on the main menu to build your request. In this example, you use the TABLES, COLUMNS, CONDITIONS, JOIN, TARGET, and DB ACCESS options.

**Step 2. Choosing the source tables**

The first step in creating the extract, after starting a End User Dialogs session, is to select the tables that contain the data you want to extract. To select these tables follow the steps shown below:

1. Select the Tables option either by pressing F4 (TABLES), or by typing 1 on the option line and pressing ENTER.

The 'Select Tables/DXTVIEWs for Extract' panel is displayed.
Figure 105. Select Tables/DXTVIEWs for Extract panel

This panel displays a list of Tables and DXTVIEWs, that you can use in your extract requests:

- Relational data stored in DB2 and SQL/DS is represented as tables, which are listed by Table Name.

- Data stored in non-relational data sources (IMS databases, VSAM, or physical sequential files) or accessed through generic data interface (GDI) exits (like remote data sources) are listed by DXTVIEW.

In this example you will extract data from two DB2 databases, and so will use two tables.

Each table or DXTVIEW is shown as a separate line on the panel, with seven descriptive fields for each table. To see the rest of the fields for a table or DXTVIEW, press F12. Press F11 (LEFT) to return to the start of the fields. A description of the contents is each field is available, and can be accessed by pressing F1 (HELP).

If there are more tables and DXTVIEWs than can be listed on a single panel, press F8 (FORWARD) to scroll forward through the list. F7 (BACKWARD) can be used to scroll backward.

2. Press F12 (RIGHT) to scroll to the right and display the description field.

3. Type an S in the SELECT column next to the descriptions for DB2ORG and DB2STAFF.
The panel should now look similar to the one shown below.

```
SELECT TABLES/DXTVIEWS FOR EXTRACT
ROW 1 OF 13

Enter an s under SELECT to select table(s) or DXTVIEW(s) you wish to extract
from. Remember to select items from the same LOCATION and of the same TYPE.
To view additional table information, scroll RIGHT.

SELECT DXTNODE
S NAME DESCRIPTION
== s DATAREFRESHER SAMPLE TABLE
== s DATAREFRESHER SAMPLE TABLE
== s DATAREFRESHER SAMPLE TABLE
== = HISAM, VIEW TO HIST
== = HISAM, VIEW TO PROJ
== = HISAM, VIEW TO STAFF
== = DEPARTMENT ORGANIZATION DATA
== = CURRENT BUSINESS PROJECTS
== = EMPLOYEE STATISTICS AND HISTORY
== = DEPARTMENT ORGANIZATION DATA
== = CURRENT BUSINESS PROJECTS

Note: Remote nodes are no longer supported.

COMMAND ==== SCROLL ==== PAGE
F1=HELP F2=SEND F3=END F4=TABLES F5=COLUMNS F6=CONDITION
F7=BACKWARD F8=FORWARD F9=JOIN F10=TARGET F11=LEFT F12=RIGHT
```

Figure 106. Select Tables/DXTVIEWS for Extract example panel

Note: Your selections are saved during this End User Dialogs session. Except for the 'Database Access' panel, pressing ENTER does not affect your input, nor does it bring up the next dialog panel.

4. Press F5 (COLUMNS) to display the 'Select Columns for Extract' panel.

Step 3. Selecting the columns

The 'Select Columns for Extract' panel should be displayed to complete this step.

```
SELECT COLUMNS FOR EXTRACT
ROW 1 of 12

Do you want to select ALL columns for extract, Yes or No? ===> N

If not, enter an s in the SELECT column to select the specific column for
extract and enter any FUNCTIONS for the columns. Press HELP for a list
of functions.

SELECT FUNCTION COLUMN TABLE CREATOR
S NAME NAME
== == DEPT DB2ORG DVR
== == DEPTNAME DB2ORG DVR
== == DIVISION DB2ORG DVR
== == LOCATION DB2ORG DVR
== == MANAGER DB2ORG DVR
== == COMM DB2STAFF DVR
== == DEPT DB2STAFF DVR
== == ID DB2STAFF DVR
== == JOB DB2STAFF DVR

COMMAND ==== SCROLL ==== PAGE
F1=HELP F2=SEND F3=END F4=TABLES F5=COLUMNS F6=CONDITION
F7=BACKWARD F8=FORWARD F9=JOIN F10=TARGET F11=LEFT F12=RIGHT
```

Figure 107. Select Columns for Extract panel
Each line on this panel corresponds to a column of DB2ORG or DB2STAFF. There are five descriptive fields for each line, press F1 (HELP) for a description of the fields.

If there are more columns than can be listed on one panel, press F8 (FORWARD) to scroll through the list. Press F7 (BACKWARD) to scroll backwards through the list.

1. To select data from every column in the source table, you would simply enter a Y on the input line at the top of the panel. In this example, you only want to extract data from four columns, so leave the value N.

2. Type an S in the Select field for the DEPT and DEPTNAMES columns (DB2ORG), and in the Select field for the ID column (DB2STAFF).

These columns contain, respectively: a department number, the name of the department, and the department member's employee number.

The panel should now show that you have selected three columns.

3. Press ENTER.

4. Press F8 (FORWARD) to scroll to the bottom of the list.

5. Type an S in the SELECT field of the SALARY column. This column contains the salary for each employee.

![SELECT COLUMNS FOR EXTRACT](image)

Do you want to select ALL columns for extract, Yes or No? ==> N

If not, enter an s in the SELECT column to select the specific column for extract and enter any FUNCTIONs for the columns. Press HELP for a list of functions.

<table>
<thead>
<tr>
<th>SELECT</th>
<th>FUNCTION</th>
<th>COLUMN</th>
<th>TABLE</th>
<th>CREATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>=&gt;</td>
<td>=&gt;</td>
<td>NAME</td>
<td>DB2STAFF</td>
<td>DVR</td>
</tr>
<tr>
<td>=&gt; s</td>
<td>=&gt;</td>
<td>SALARY</td>
<td>DB2STAFF</td>
<td>DVR</td>
</tr>
<tr>
<td>=&gt;</td>
<td>=&gt;</td>
<td>YEARS</td>
<td>DB2STAFF</td>
<td>DVR</td>
</tr>
</tbody>
</table>

***END OF DATA***

*Figure 108. Select Columns for Extract panel*

6. Press ENTER.

You can now specify the data you want, by entering built-in functions in the appropriate columns. In this example, you have to count the number of employees in each department. Each employee has a distinct employee number so you can count the employee numbers in each department to tell you how many employees are in each department.

7. Press F7 (BACKWARD) to return to the first list of columns.

8. Type COUNT under the function column next to ID. This tells DataRefresher that you want to count the IDs. Several other Functions are available, with DataRefresher, some of these are described later in this chapter. However, you can use the Help key to display information about the Functions.

9. Press ENTER.

10. Press F8 (FORWARD) to scroll to the next panel.
11. You can also average the salaries of each department’s members. Type AVG beside Salary, typing over the GROUP function which is displayed in the Function column.

When a function is specified for selected columns, all remaining selected columns must have GROUP in the FUNCTION field. Without GROUP, your output would contain several rows of data for the same department, and you want one row of salary information per department. GROUP summarizes the data for you, and creates one row for each different department. This value is automatically supplied by DataRefresher.

12. Press ENTER.

13. Press F7 (BACKWARD) to scroll back to the preceding panel. GROUP is now displayed in the Function column next to the DEPTNAME and DEPT column names.

Note: For descriptions of other built-in functions you can use in the FUNCTION column, press F1 (HELP).

The two panels should now be similar to the ones shown below.

```
SELECT ANSWER 1 OF 16

Do you wish to select ALL columns for extract, Yes or No? === N

If not, enter an S in the SELECT column to select specific columns for extract and enter any FUNCTIONs for the columns. Press HELP for a list of functions.

SELECT FUNCTION COLUMN TABLE CREATOR

=> => DEPT DB2ORG DVR
=> S => GROUP DEPT DB2ORG DVR
=> => DEPTNAME DB2ORG DVR
=> S => GROUP DEPTNAME DB2ORG DVR
=> => DIVISION DB2ORG DVR
=> => LOCATION DB2ORG DVR
=> => MANAGER DB2ORG DVR
=> => COMM DB2STAFF DVR
=> => DEPT DB2STAFF DVR

COMMAND === F2=SEND F3=END F4=TABLES F5=FUNCTIONS F6=CONDITION
F7=BACKWARD F8=FORWARD F9=JOIN F10=TARGET F11=EXT LIST F12=ACCESS

Figure 109. Select Columns and Specify Functions left panel
```
Do you want to select ALL columns for extract, Yes or No?  ——> N

If not, enter an s in the SELECT column to select specific columns for extract and enter any FUNCTIONs for the columns. Press HELP for a list of functions.

<table>
<thead>
<tr>
<th>SELECT</th>
<th>FUNCTION</th>
<th>COLUMN</th>
<th>TABLE</th>
<th>CREATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>=&gt;</td>
<td>=&gt;</td>
<td>ID</td>
<td>DB2STAFF</td>
<td>DVR</td>
</tr>
<tr>
<td>=&gt;</td>
<td>=&gt;</td>
<td>COUNT</td>
<td>DB2STAFF</td>
<td>DVR</td>
</tr>
<tr>
<td>=&gt;</td>
<td>=&gt;</td>
<td>JOB</td>
<td>DB2STAFF</td>
<td>DVR</td>
</tr>
<tr>
<td>=&gt;</td>
<td>=&gt;</td>
<td>NAME</td>
<td>DB2STAFF</td>
<td>DVR</td>
</tr>
<tr>
<td>=&gt;</td>
<td>=&gt;</td>
<td>SALARY</td>
<td>DB2STAFF</td>
<td>DVR</td>
</tr>
<tr>
<td>=&gt;</td>
<td>=&gt;</td>
<td>AVG</td>
<td>DB2STAFF</td>
<td>DVR</td>
</tr>
<tr>
<td>=&gt;</td>
<td>=&gt;</td>
<td>YEARS</td>
<td>DB2STAFF</td>
<td>DVR</td>
</tr>
</tbody>
</table>

Figure 110. Select Columns and Specify Functions right panel

**Note:** When you press ENTER after entering built-in functions, DataRefresher produces a duplicate row, allowing you to specify multiple functions for the same column. For example, after you enter this built-in function:

```plaintext
=> => AVG SALARY DB2STAFF DVR
```

The following is displayed:

```plaintext
=> => SALARY DB2STAFF DVR
=> => AVG SALARY DB2STAFF DVR
```

14. Press F6 (CONDITION) to display the 'Specify Conditions' panel.
Step 4. Specifying conditions

The 'Specify Conditions' panel should be displayed to complete this step.

---

**SPECIFY CONDITIONS**

Specify the CONDITION field as operator followed by value(s). You may also use the lines below for free formatted conditions. Press HELP key for syntax.

---

<table>
<thead>
<tr>
<th>COLUMN NAME</th>
<th>COLUMN TYPE</th>
<th>COLUMN LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEPT</td>
<td>SMALLINT</td>
<td></td>
</tr>
<tr>
<td>DEPTNAME</td>
<td>VARCHAR</td>
<td>14</td>
</tr>
<tr>
<td>DIVISION</td>
<td>VARCHAR</td>
<td>10</td>
</tr>
</tbody>
</table>

---

**Figure 111. Specify Conditions panel**

On the previous panel, the 'Select Columns and Specify Functions' panel, you selected a subset of the columns from the two DB2 tables, that data is to be extracted from.

Now you select the rows from which the data is to be extracted. Each row in the DB2ORG table contains information about each of the departments, while each row in DB2STAFF contains information about one employee who is employed in a department. Because you only want to know the number of employees in departments numbered 15 or 20, you only want DataRefresher to extract data from rows in the DB2ORG table with 15 or 20 in their DEPT column. Therefore, you use the 'Specify Conditions' panel to specify conditions that determine which row DataRefresher extracts.

Not all the columns in both tables are shown on the panel. You can use F8 (FORWARD) and F7 (BACKWARD) to scroll through the list of columns.

To specify a condition which must be matched before DataRefresher will extract data from a row, you can use the two free-format lines at the top of the panel, or the Condition field for a particular column.

1. To specify the condition for this example, where you only want data from rows for department 15 and 20, enter `DEPT = 15 OR DEPT = 20` on a Free Format line.
Specify the CONDITION field as operator followed by value(s). You may also use the lines below for free formatted conditions. Press HELP key for syntax.

```sql
==> DEPT = 15 OR DEPT = 20
==>
```

<table>
<thead>
<tr>
<th>COLUMN NAME</th>
<th>COLUMN TYPE</th>
<th>COLUMN LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEPT CONDITION: ==&gt;</td>
<td>SMALLINT</td>
<td></td>
</tr>
<tr>
<td>DEPTNAME CONDITION: ==&gt;</td>
<td>VARCHAR</td>
<td>14</td>
</tr>
<tr>
<td>DIVISION CONDITION: ==&gt;</td>
<td>VARCHAR</td>
<td>10</td>
</tr>
</tbody>
</table>

Figure 112. Specifying conditions using the Specify Conditions panel

Step 5. Joining two source columns

The 'Specify Joins' panel should be displayed to complete this step. The panel lists all the selected columns from the source tables. Not all the columns and fields in both tables are shown on the panel. You can use F8 (FORWARD) and F7 (BACKWARD) to scroll through the list of columns, and F12 (RIGHT) and F11 (LEFT) to scroll through the list of fields.

```sql
When multiple tables are selected for extract, the columns to be joined should be specified. Enter the number from the COLUMN NUMBER field for one of the columns into the JOIN COLUMN field for the other column. For additional join information, scroll RIGHT.
```

<table>
<thead>
<tr>
<th>COLUMN NUMBER</th>
<th>JOIN COLUMN NAME</th>
<th>TABLE/DXTVIEW NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>001 =&gt; DEPT</td>
<td></td>
<td>DB2ORG</td>
</tr>
<tr>
<td>002 =&gt; DEPTNAME</td>
<td></td>
<td>DB2ORG</td>
</tr>
<tr>
<td>003 =&gt; DIVISION</td>
<td></td>
<td>DB2ORG</td>
</tr>
<tr>
<td>004 =&gt; LOCATION</td>
<td></td>
<td>DB2ORG</td>
</tr>
<tr>
<td>005 =&gt; MANAGER</td>
<td></td>
<td>DB2ORG</td>
</tr>
<tr>
<td>006 =&gt; COMM</td>
<td></td>
<td>DB2STAFF</td>
</tr>
<tr>
<td>007 =&gt; DEPT</td>
<td></td>
<td>DB2STAFF</td>
</tr>
<tr>
<td>008 =&gt; ID</td>
<td></td>
<td>DB2STAFF</td>
</tr>
<tr>
<td>009 =&gt; JOB</td>
<td></td>
<td>DB2STAFF</td>
</tr>
<tr>
<td>010 =&gt; NAME</td>
<td></td>
<td>DB2STAFF</td>
</tr>
</tbody>
</table>

Figure 113. Specify Joins panel
In this step you tell DataRefresher which columns to join from the DB2ORG and DB2STAFF tables. In this example you use the department number to identify the columns to be joined.

DataRefresher passes that information to DB2, which carries out the join. DB2 concatenates the information for selected the columns, that is the row information, then examines the concatenated row, using the conditions specified on the Specify Conditions and 'Specify Join' panels. Any row not meeting the specified conditions is eliminated. If the values in both the DB2ORG and DB2STAFF Dept columns are the same and meet the specified conditions, the two rows are concatenated and added to the JOIN table.

To join the two Dept columns – Dept-001 and Dept-007 – which contain like information, type the column number of one of the columns in the opposite column's Join Column field. For this example, type 007 in the Join Column field for column number 001, as shown below.

```
<table>
<thead>
<tr>
<th>COLUMN JOIN COLUMN NUMBER COLUMN NAME</th>
<th>TABLE/DXTVIEW NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>001 =&gt; 007 DEPT</td>
<td>DB2ORG</td>
</tr>
<tr>
<td>002 =&gt; DEPTNAME</td>
<td>DB2ORG</td>
</tr>
<tr>
<td>003 =&gt; DIVISION</td>
<td>DB2ORG</td>
</tr>
<tr>
<td>004 =&gt; LOCATION</td>
<td>DB2ORG</td>
</tr>
<tr>
<td>005 =&gt; MANAGER</td>
<td>DB2ORG</td>
</tr>
<tr>
<td>006 =&gt; COMM</td>
<td>DB2STAFF</td>
</tr>
<tr>
<td>007 =&gt; DEPT</td>
<td>DB2STAFF</td>
</tr>
<tr>
<td>008 =&gt; ID</td>
<td>DB2STAFF</td>
</tr>
<tr>
<td>009 =&gt; JOB</td>
<td>DB2STAFF</td>
</tr>
</tbody>
</table>
```

Figure 114. Joining panels using the Specify Joins panel

Press F10 (TARGET) to display the 'Name Target for Extract Output' panel.
Step 6. Identifying the target table

The 'Name Target for Extract Output' panel should be displayed to complete this step.

Figure 115. Name Target for Extract Output left panel

To complete this step, the last for this task, follow the steps below. For a description of the fields on this panel, press F1 (HELP).

1. Type your entries in the following fields in the top half of the panel:
   - **Target Nickname**: This is the DataRefresher nickname for your target node. For this example SF01 is used.
   - **Target Table Name**: Type the name of the target table in this field. For this example, use DEPTSAL.
   - **Table Qualifier**: Type the prefix of your target table in this field. For this example, use your user ID.
   - **Target Table Option**: Type C, to show that you want to create a SQL/DS table.

   **Note**: If you do not know what DataRefresher nickname you should specify, type a question mark (?) in the Target Nickname field and press ENTER. The 'List of Nicknames' panel is displayed. This panel contains a complete list of nicknames which you can use. To select a nickname, type an S in the SEL column next to the nickname, and press ENTER, to return to the 'Name Target for Extract Output' panel.

2. To change the names of the target columns from their defaults, specify new target column names in the Target Column Name field overwriting the DataRefresher defaults.

   This step is optional, you can use the target column names that DataRefresher provides.
3. Press F12 (RIGHT) to scroll to the right and display the remaining fields for the columns.

<table>
<thead>
<tr>
<th>NAME TARGET FOR EXTRACT OUTPUT</th>
<th>Row 1 of 16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Nickname =&gt; SFO1</td>
<td>Specify nickname of target system</td>
</tr>
<tr>
<td>Target Table Name =&gt; DEPTSAL</td>
<td>Specify for DB2, SQL/DS &amp; IXF</td>
</tr>
<tr>
<td>Table Qualifier =&gt; YOURID</td>
<td>Prefix for target table name</td>
</tr>
</tbody>
</table>

Specify when Target is SQL/DS or DB2:
Target Table Option => C
C / R / blank

Change the TARGET COLUMN NAMES where appropriate.

<table>
<thead>
<tr>
<th>INPUT COLUMN</th>
<th>ORDER SEQUENCE</th>
<th>OUTPUT ORDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>(1 - 8)</td>
<td>(ASC/DESC)</td>
</tr>
<tr>
<td>DEPT</td>
<td>=&gt;</td>
<td>=&gt;</td>
</tr>
<tr>
<td>DEPTNAME</td>
<td>=&gt;</td>
<td>=&gt;</td>
</tr>
<tr>
<td>ID</td>
<td>=&gt;</td>
<td>=&gt;</td>
</tr>
<tr>
<td>SALARY</td>
<td>=&gt;</td>
<td>=&gt;</td>
</tr>
</tbody>
</table>

********************************** BOTTOM OF DATA **********************************

Figure 116. Name Target for Extract Output right panel

If you had not used built-in functions, you could use the Order Sequence field to specify how the rows would be arranged. When you use built-in functions, you cannot use this field to order your data because it can only be used with the columns from the source data.

4. Press F11 (LEFT) to return to the left half of the panel.

You have now completed this task, and should continue with the task described in “Task 2. Providing database access information” on page 189.
Task 2. Providing database access information

In this task you provide DataRefresher with the database access information it needs to read the source data. This information is entered on the 'Database Access' panel. To display this panel, press F12 (ACCESS).

<table>
<thead>
<tr>
<th>DATABASE ACCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Source information:</strong></td>
</tr>
<tr>
<td>Location ====&gt; NYC1 Location of table(s) to be extracted.</td>
</tr>
<tr>
<td>User ID ====&gt; YOURID User ID to access source table(s).</td>
</tr>
<tr>
<td>Password ====&gt; Access password of above User ID.</td>
</tr>
<tr>
<td>VM Execution ID ===&gt; Userid to execute VM extract request.</td>
</tr>
</tbody>
</table>

**Target information when source is SQL/DS:**

| User ID ====> User ID to execute the job which processes the extracted output. |
| Network ID ====> Net ID to route extracted output. |
| Target Node ID ====> SFOVM1 Location for extracted output. |

Press ENTER to save the required information.

**COMMAND ====>**

F1=HELP F2=SEND F3=END F4=TABLES F5=COLUMNS F6=CONDITION
F7=BACKWARD F8=FORWARD F9=JOIN F10=TARGET F11=EXT LIST F12=ACCESS

*Figure 117. Database Access panel*

**Note:** The 'Database Access' panel is the only panel that saves your entries when you press ENTER.

When the panel is displayed it contains default entries for the Location, User ID, and Target Node ID fields.

1. If your User ID is different from the displayed User ID, type your User ID in the field.

2. Type your password, for the specified User ID, in the Password field. Note that the password is not displayed when you type it.

3. When you extract data from:

   **SQL/DS table** You must type in the Target User ID and the Target Node ID.

   **Other sources** The Target User ID and Target Node ID are provided in the JCS.

4. Press ENTER.

   A message is displayed informing you that the database access information has been added to the Access Table.

**Notes:**

1. You should create an entry in DataRefresher for each source node from which you want to extract data. DataRefresher saves database access information separately from the extract requests, so you do not need to repeat the information for which particular source node.

2. If you select a table from the same node as your source table when you build request, DataRefresher displays the database access details for this table in
the 'Database Access' panel. You may change the entries by entering different information on the panel. What you type on the panel replaces the information already stored in your Database Access table.

3. You can add an entry for any node to the 'Database Access' panel, at any time. For example, to add database access information for another location because you need to extract data from it later, provide the access information now. You don't have to wait until you are building the extract request to add an access entry.

---

**Task 3. Saving the extract request**

When you have completed the previous tasks, and specified the information for the Tables, Columns, Conditions, Target, and Database Access panels, you can add your request to the dialogs library. Once the request has been stored it can be used it in future End User Dialogs sessions.

To save the extract you have to give it a name, using the SAVE command. The name you use must be unique in the dialogs library, and it can be no more than 18 characters long.

Before saving the request, check whether a name is unique by pressing F11 (EXT LIST) Type the command DISPLAY on the command line before saving the request. The 'Display Extract Request' panel is displayed. This panel lists the extract requests that already exist in your library. To return to the previous panel, press F3 (END)

**Note:** If you are in an object sharing session, only names contained in the common object library are shown in the panel.

You can add a description when you save your extract request, although DataRefresher does not require one. A description may remind you what your extract request actually produces. The description is included in the SAVE command, and must be enclosed in single quotes and entered after the name of the extract request. It can be up to 31 characters long. For example to save the extract that you created in this chapter, and call it EXTTTEST with the description *Avg salaries/Dept 15 and 20*, you would enter the following command on the command line:

```
SAVE AS EXTTTEST (DESC = 'Avg salaries/Dept 15 and 20')
```

For more information about the SAVE command, see the *DataRefresher Command Reference*

After you entered the SAVE command, you might want to verify that DataRefresher saved your request. Press F11 (EXT LIST) or type DISPLAY on the command line to look at the 'Display Extract Request' panel. The request you should be included in the list of extract requests.

**Note:** You can use the 'Display Extract Request' panel, to add or change a description of an extract request. Type the new or changed description in the Description field and press ENTER.

If you specified object sharing when you logged on to the End User Dialogs, your saved request goes to a shared library containing extract requests. You must have WRITE access to save extract requests to the shared library.
If you do not have WRITE access to the shared library, or if you are not accessing the shared library in this End User Dialogs session, your saved extract request now resides in your personal library of extract requests.

Next time, you do not have to repeat the steps to build this extract request. Just press F11 or type DISPLAY, then select the extract request you want by typing S in the Select field next to the Request Name.

**Task 4. Sending the extract request**

At this point, you have:

- Built an extract request.
- Provided DataRefresher with the access information.
- Saved the extract request.

Also, your administrator has created the JCL and JCS you need to use for this extract request. These JCL and JCS files must be specified on your 'Profile Review' panel. The information specified on this panel is saved with your request. If you are unsure whether the JCL and JCS files specified in your profile are correct, check the 'Profile Review' panel and consult your DataRefresher administrator.

**Note:** For this example, you need to change the Target Type field on the Profile Review panel to SQL/DS. See Chapter 21, "JCL/JCS files and End User Dialogs requests" on page 193 for more details.

To send a request for processing:

1. Enter the SEND command followed by the name of the request. For example:
   
   SEND EXTTEST

   **Note:** You can enter the command without specifying the name of a request. If you do this DataRefresher will send the currently active request for processing. You make a request active by creating it, or selecting it from the 'Display Extract Request' panel.

   The 'Send Confirmation' panel is displayed.

   ![Send Confirmation panel](image)

   **Figure 118. Send Confirmation panel**

2. To confirm that the request is to be sent for processing press ENTER.
The extract request is sent for processing, and you can:

- Start another End User Dialogs session:
  1. Press F4 (TABLES) to return to the 'Select Tables/DXTVIEWS for Extract' panel.
  2. Type RESET on the command line and press ENTER. The RESET command clears all selections from the previous request.
  3. Select the table or DXTVIEW for another dialog.

- Return to the ISPF/PDF Primary Option Menu to start a Administrative Dialogs session:
  1. Press F3 (END) to return to the End User Dialogs main menu.
  2. Press F3 (END) to return to the ISPF Primary Option menu.
  3. Select the DataRefresher Administrative Dialogs option on the menu

- Exit the DataRefresher dialogs:
  1. Press F3 (END) to return to the End User Dialogs main menu.
  2. Press F3 (END) to return to the ISPF Primary Option menu.
  3. Choose another option on the ISPF/PDF Primary Option Menu, or exit ISPF by pressing F3 (END).
Chapter 21. JCL/JCS files and End User Dialogs requests

This chapter explains how JCL or JCS files are associated with extract requests when sending a request in the End User Dialogs. It also describes the priorities set by DataRefresher to associate the proper JCL or JCS file with the request.

To send your extract request to DataRefresher, you need to use the appropriate JCL. To control the extracted data and move it to the correct target in the proper format, you need to use the appropriate JCS. When you use the SAVE or SEND command with the End User Dialogs, DataRefresher associates your request with the JCL and JCS named in your profile, or specified through a source or target nickname.

The JCL and JCS files you need must be in your dialogs library or, if you are using object sharing, the shared library. Your administrator sets up your dialogs or shared library to include the necessary JCL and JCS files, then specifies these files for association with your End User Dialogs requests through:

- Your user profile (built in Administrative Dialogs)
- An entry in the nickname table that associates a specific source node with a specific JCL file

You or your administrator can also associate a JCS file with your extract request on the 'Name Target For Extract Output' panel. See Chapter 14, “Changing the default profile” on page 127 for information about building the End User Dialogs profile.

Using the Profile Review panel
From the End User Dialogs main menu, type 7 in the Option field and press Enter, to display the 'Profile Review' panel. It contains profile information values that have been entered by your administrator for your End User Dialogs session.
PROFILE REVIEW

COMMAND ===>

Type 'Y' to permanently save in profile, when you press END===

JCL Name ===> JCL to execute request.
Control Deck Output ===> 1=JCS, 2=Phy Seq, 3=None.
Extract Output ===> 1=JCS, 2=Phy Seq, 3=Both.
If JCS, JCS Name ===> JCS to process data.
If Phy Seq, ddbname ===> ddbname in execution JCL.
or, MVS Dynamic Allocation attributes
Data Set Name ===>
Unit ===> Volume ===
Password ===> Blocking Factor ===
or, CMS Dynamic Allocation attributes
CMS File Name ===> Type ===> Mode ===
Output Limit ===> Specify 1 to 10000000 or None.
Priority ===> Specify 0 to 255.
Format ===> 1=EBDIC, 2=Source.
Target Type ===> D=DB2, S=SQL/DS, I=IXF, N=N/A.

COMMAND ===>
F1=HELP F2=SEND F3=END F4=TABLES F5=COLUMNS F6=CONDITION
F7=BACKWARD F8=FORWARD F9=JOIN F10=TARGET F11=EXT LIST F12=ACCESS

Figure 119. Profile Review panel

Each field of the Profile Review panel represents a value from either your DataRefresher profile, or the extract request that is currently active. Certain fields may have defaults set. For example, Target Type defaults to DB2, and JCS Name defaults to the JCS that runs the DB2 load facility.

You can change information on the Profile Review panel before you SAVE or SEND your next request. To save the changes permanently, type Y in the field at the top of the panel and press END (F3). To save the changes for this End User Dialogs session, press END (F3).

DataRefresher makes associations based on the following priorities:

1. The nickname table is searched for a JCL file corresponding to a specific nickname.
2. The profile value is used.

If you want to override these values, use the 'Profile Review' panel. Every time you save or send a request, a JCL or JCS file is associated with an extract request based on what is displayed in the 'Profile Review' panel.

If the JCL or JCS file does not exist, either in the user's personal DataRefresher dialogs library or in the DataRefresher dialogs shared objects library, your extract request will fail when you send it.

If the file exists but is not specified in your profile, use the 'Profile Review' panel to complete the appropriate fields.

When you send the request, DataRefresher dialogs submit a job that includes your extract request and the appropriate JCL and JCS files to the UIM or REM using the DVREXSND EXEC (VM) or a TSO SUBMIT (MVS).
Part 4. Appendixes, terms and abbreviations
Appendix A. Data set and file definitions

The following allocations may vary according to the user's needs. For example, as an administrator, you should probably have a larger DVRJEDIE than the average DataRefresher user because you are working with more JCL and data descriptions.

MVS common data set definitions

All of these data sets are libraries (partitioned data sets).

<table>
<thead>
<tr>
<th>Accessed by dd name</th>
<th>Default data set name</th>
<th>Function/content</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPPLIB</td>
<td>DVR110.DVRPLIBE</td>
<td>Contains the DataRefresher dialogs prompt and HELP panels.</td>
</tr>
<tr>
<td>ISPLLIB</td>
<td>DVR110.DVRLOAD</td>
<td>Contains the DataRefresher dialogs load library.</td>
</tr>
<tr>
<td>ISPLMLIB</td>
<td>DVR110.DVRMLIBE</td>
<td>Contains DataRefresher dialogs messages.</td>
</tr>
</tbody>
</table>
| ISPSLIB             | DVR110.DVRSLIB        | Contains ISPF skeletons that are used when DataRefresher dialogs:  
|                     |                       | • Imbed a request in a JCL file that you want to send  
|                     |                       | • Build a request from prompt-form request entries |
| DVREUADD            | DVR110.DVRTADME       | Contains the End User Dialogs command table and the master index, end user, and nickname tables built by DataRefresher dialogs during End User Dialogs administration.  
| ISPTLIB             | DVR110.DVRTADME       | Note: Because the size of the master index table depends on the particular needs of your site, you may want to evaluate this data set before allocating space. |
| DVRSTASL            | DVR110.DVRTLIB        | Contains the object sharing library. |

MVS user-specific data set definitions

<table>
<thead>
<tr>
<th>Accessed by dd name</th>
<th>Default data set name</th>
<th>Function/content</th>
<th>Recommended storage (3350)</th>
</tr>
</thead>
</table>
| ISPPROF             | userid.DVR110. DVRTPROF | Contains ISPF profile information. userid is the user's user ID. Depending on your system configuration, this data set may have another name.  
|                     |                       | Note: If the user you are enrolling already has an ISPF profile, do not allocate this data set. | (TRK,(5,5,30)) |
| ISPTLIB             | userid.DVR110. DVRTLIB  | Contains the ISPF tables for the user's extract requests, information about the sends of all extract requests, and other DataRefresher dialogs information. userid is the user's user ID. | (CYL,(1,1,30)) |
| ISPTABL             | userid.DVR110. DVRTLIB  | Contains the user's edited extract requests, data description requests, and JCL files. The data set name shown here may be changed to conform to your own naming conventions. It is recommended, but not mandatory, that a user ID (userid) is used as a first-level qualifier for the data set name. | (CYL,(3,1,30)) |

© Copyright IBM Corp. 1986, 1994
### VM common file definitions

<table>
<thead>
<tr>
<th>Accessed by dd name</th>
<th>Default data set name</th>
<th>Function/content</th>
<th>Recommended storage (3350)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVRDIMEX</td>
<td>userid.DVR110. DVRIMEX</td>
<td>Contains the user's edited extract requests, data description requests, and any JCL files that the user has exported or imported in sharing objects with other users. userid is the user's user ID.</td>
<td>(CYL,1,1,30)</td>
</tr>
</tbody>
</table>

### VM user-specific file definitions

<table>
<thead>
<tr>
<th>Accessed by dd name</th>
<th>Default data set name</th>
<th>Function/content</th>
<th>Recommended storage (3350)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPPROF</td>
<td>userid MACLIB</td>
<td>This file contains ISPF profile information. Depending on your system configuration, this file may have another name. userid is the user's user ID.</td>
<td>1 cylinder</td>
</tr>
<tr>
<td>Note: If the user you are enrolling already has an ISPF profile, do not allocate this file.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISPTLIB ISPTABL</td>
<td>DVRLIBE MACLIB</td>
<td>Contains the ISPF tables for the user's extract requests, information about the sends of all extract requests, and other DataRefresher dialogs information.</td>
<td>1 cylinder</td>
</tr>
<tr>
<td>DVRJDJEDI ISPSTLIB</td>
<td>DVRJEDIE MACLIB</td>
<td>Contains the user's edited extract requests, data description requests, and JCL files. The file name shown here may be changed to conform to your own naming conventions. The file type is still MACLIB (VM).</td>
<td>3 cylinders</td>
</tr>
<tr>
<td>DVRDIMEX</td>
<td>DVRIMEXE MACLIB</td>
<td>Contains the user's edited extract requests, data description requests, and any JCL files that the user has exported or imports in sharing objects with other users.</td>
<td>1 cylinder</td>
</tr>
</tbody>
</table>
Terms and abbreviations

This glossary defines terms and abbreviations as they are used in the DataRefresher library. Entries often include further information about how the term applies specifically to DataRefresher.

**abend.** Abnormal end of a task.

**access control list (ACL).** Defines the access rights to the associated data source in DataRefresher.

**access level.** The level of authority a user has when using a protected DataRefresher resource or command set.

**Accounting exit routine.** A user-written routine used for charging resources to individual users. This routine is started at the beginning and the end of the extract request execution cycle. It can be used to change output limits, change priorities of the DEM, and account for the resources used by the DEM on behalf of individual extract requests. This routine can be coded in Assembler, PL/I, or COBOL.

**ACL.** access control list.

**Administrative Dialogs.** A series of menus and displays that help a user create and submit data descriptions and extract requests, maintain the profiles for DataRefresher Dialogs and JCL for submitting data descriptions and extract requests, and administer End User Dialogs.

**Advanced Program-to-Program Communications (APPC).** The communication protocol (LU 6.2) that is used by DataRefresher.

**alias.** An alternate name for a member of a partitioned data set or for a name of a field described in a data description.

**APAR.** authorized program analysis report.

**APPC.** Advanced Program-to-Program Communications.

**Application System (AS).** An IBM* integrated decision support program that helps provide business planning, graphics, project control and management, statistical data analysis, and other functions.

**AS.** Application System.

**asynchronous.** Occurring without a regular or predictable time relationship.

**ASCII.** ANSI Standard Code for Information Interchange.

**authorized program analysis report (APAR).** A report of a problem caused by a suspected defect in a current, unaltered release of a program.

**batch message processing (BMP).** An IMS/VS region where batch message processing occurs.

**BMP.** batch message processing.

**Boolean expression.** In DataRefresher, a conditional expression that evaluates to true or false to determine whether a particular unit of data is extracted. The expression may contain multiple conditions connected by the logical operators AND, OR, and NOT.

**card-image input.** Input that simulates punched card input (80 columns per record).

**CCU.** Consistency Check Utility - a DataPropagator NonRelational feature used with the DataRefresher UIM when using a DataPropagator NonRelational map capture exit in the DataRefresher SUBMIT command.

**CEEPPI.** Common Execution Environment Pre-Initialized Program Interface.

**child segment.** In a database, a segment that lies immediately below its parent segment. A child segment has only one parent segment.

**CLIST.** command list.

**CMS.** conversational monitor system.

**code page.** An assignment of graphic characters and control function meanings to all code points.

**code point.** A 1-byte code representing one of 256 potential characters.

**command list (CLIST).** A data set or a member of a partitioned data set containing TSO commands that run sequentially in response to the EXEC command.
command string. A language construct that represents one step in a sequence of steps that produce a DataRefresher command.

containing segment. A parent segment that contains one or more internal segments.

control blocks. Storage areas used to hold control information.

conversation. An exclusive use of an LU-to-LU session by two transaction programs using the APPC. This is a short logical connection that lasts only for the duration of one complete transaction. (Contrast with session).

conversational monitor system (CMS). A virtual machine operating system that provides general interactive time sharing, problem solving, and program development capabilities.

DAP. Dictionary Access Program.

database format. The format of the data prior to any segment preprocessing or data exit manipulation. (Contrast with FDTLIB format).

database manager. A program that controls the user's data, ensuring security and data integrity in a multiple user environment. Examples include Information Management System/Virtual Storage (IMS/VS), IBM DATABASE 2 (DB2), and Structured Query Language/Data System (SQL/DS).

Database Management System (DBMS). A software system that controls the creation, organization, and modification of a database and access to the data stored within it.

data definition (DD) statement. A job control statement that describes the data sets associated with a specific job step.

data description. The description of a file, IMS/VS DL/I database, or any data source accessed by a user-written generic data interface (GDI) exit.

Data Dictionary. The IBM OS/VS DB/DC Data Dictionary is a central repository of information about data such as names, meaning, relationships to other data, origin, usage, and format.

data entry database (DED8). An IMS/VS Fast Path database used to provide efficient access to large volumes of detailed data. Each DEDB can be partitioned, or divided into multiple "areas" for ease of access.

Data exit routine. A user-written routine to provide data verification. The routine can be used to process each record in source files and each segment in source databases. This routine can be coded in assembler, PL/I, or COBOL.

DataRefresher. An IBM program that extracts data from a source database or file and formats it for a target database or file.

Data Extract Manager (DEM). The DataRefresher program that extracts data from a VSAM file, a physical sequential file, an IMS/VS DL/I database, or any data source accessed by a generic data interface (GDI) exit.

data facility sort (DFSORT). An IBM product that works in conjunction with DataRefresher or other database products to sort data as it is being processed.

Data Language 1 (DL/I). The database management language for IMS/VS.

data propagation. The process of applying the changes to one set of data to the copy of that data in another database system.

Data Reformat Utility (DRU). This utility recombines 80-character record segments (used to transmit data from one system to another) into logical records.

Date/Time Conversion exit routine. A user-written routine to convert date/time data to ISO format. The routine can be used to process each field containing date/time data. It can be coded in assembler, PL/I, or COBOL.

DBCS. double-byte character set.

DBMS. Database Management System.

DB2. IBM DATABASE 2.

DD. data definition.

DED8. data entry database.

DEM. Data Extract Manager.

DEM data source. The source data from which the DEM extracts data. (For example, an IMS/VS DL/I database, a VSAM or physical sequential data set, or data accessed by a user-written generic data interface (GDI) exit.)

DFSORT. data facility sort.

Dictionary Access Program (DAP). The DataRefresher program that generates descriptions of the files and nonrelational databases from which users extract data. The descriptions are taken from existing definitions in the IBM Data Dictionary.

DL/I. Data Language I.
double-byte character set (DBCS). A set of characters in which each character occupies 2 bytes. Languages such as Japanese, Chinese, and Korean that contain more symbols than can be represented by 256 code points require double-byte character sets. Entering, displaying, and printing DBCS characters requires special hardware and software support.

DRT. Data Reformat Utility.

Data Refresher dialogs. The Data Refresher programs that help users build and send data descriptions (Administrative Dialogs) and extract requests (End User Dialogs and Administrative Dialogs).

DXTFILE. A Data Refresher object stored in the FDTLIB that describes a SAM, VSAM, physical sequential, or other type of data set available by using the generic data interface.

DXTFILE description. Describes a physical sequential file, VSAM file, or any other data source accessed by a generic data interface (GDI) exit. (Describes the file organization and selected fields.)

DXTPCB. Data Refresher Program Communication Block.

Data Refresher Program Communication Block (DXTPCB). A Data Refresher object stored in the FDTLIB that describes an IMS/VSE PCB to Data Refresher.

Data Refresher Program Specification Block (DXTPSB). A Data Refresher object stored in the FDTLIB that describes an IMS/VSE PSB to Data Refresher.

DXTPSB. Data Refresher Program Specification Block.

DXTPSB description. For a given PCB, a description of the fields of interest to the user and the segments where they exist. Also included in the description of a segment are its length, its format (variable or fixed length), and the name of its parent. Included in the description for a field are its origin in its segment, its length, and its data characteristics.

Data Refresher user data type. Data in a format that Data Refresher does not directly support. A user-written conversion exit is required to convert data from the unsupported format into a format that Data Refresher supports.

Data Refresher user data type description. A description of the user-defined format to Data Refresher.

DXTVIEW. A Data Refresher object stored in the FDTLIB that defines which fields or segments the user may access. A DXTVIEW describes only those segments and fields in a single path of the hierarchy that the user can retrieve.

DXTVIEW description. Defines a DXTVIEW for a physical sequential or VSAM file, an IMS/VSE DL/I database, or a data source accessed by a user-written generic data interface (GDI) exit.

EAR. Exit Address Routine.

EBCDIC. extended binary-coded decimal interchange code.

ECF. Enhanced Connectivity Facilities.

End User Dialogs. The Data Refresher program that lets a user build and submit extract requests through a series of panels.

end user table. A Data Refresher table generated to keep track of user IDs between systems.

Enhanced Connectivity Facilities (ECF). A set of programs designed to connect personal computers with host computers so that many host services and resources become available to personal computer users and application programmers.

entry-sequenced data set (ESDS). In VSAM, a file whose records are ordered by time of entry into the data set, and whose relative byte addresses cannot change. Records are retrieved and stored by sequential access, and new records are added at the end of the data set.

ESDS. entry-sequenced data set.

EXEC. A program consisting of a set of CP and CMS commands.

EXEC statement. An instruction within JCL/CMS that identifies the program to be run.

exit address routine. A user-written routine that, given the name of a GDI exit routine by Data Refresher, returns the current address of that GDI exit routine to Data Refresher.

exit routine. A user-written Data Refresher program. Data Refresher passes control to the exit routine for specialized processing.

EXTLIB. extract request library.

extended binary-coded decimal interchange code (EBCDIC). A coded character set consisting of 8-bit coded characters.

extract request. A request to extract data from a source accessible by Data Refresher. Requires use of the SUBMIT and EXTRACT commands.
extract request library (EXTLIB). A VSAM key-sequence data set (KSDS) that holds extract requests.

Fast Path. The IMS/VS function that supports applications requiring data availability and fast processing of simple data structures. Although Fast Path has its own databases and message processing, it is an integral part of IMS/VS.

FDTLIB. file description table library.

FDTLIB format. The format of a segment after a data exit transforms it, but prior to manipulation by any user data type exit during the extraction process. (Contrast with database format).

FDTLIB Migration Utility. A program used to migrate data descriptions from all prior releases of FDTLIB to the current release of FDTLIB.

field definition time. The time at which a user data type is specified in a FIELD statement of a CREATE DXTFILE or CREATE DXTPSB command.

file description table library (FDTLIB). A VSAM key-sequence data set (KSDS) that holds the descriptions of all databases, files, and DataRefresher views available to the DEM and UIM.

file space. Used by shared file system (SFS), a logical space where a user's files are kept.

file pool. A set of minidisks managed by SFS.

fixed-length record. A record having the same length as all other records with which it is logically or physically associated. (Contrast with variable-length record)

full-screen editing. Editing at a display terminal which displays an entire screen of data at once and in which the user can access data through commands or by positioning a cursor.

GDI. generic data interface.

GDI Record exit routine. A user-written program that can access a non-IBM DBMS or data source that does not have an SQL interface, extract from a self-defining file, such as an IXF file, or join data from diverse data sources. DataRefresher does not pass SELECT statements to a GDI Record exit. However, DataRefresher can pass key values to a GDI Record exit. (Contrast with GDI Select exit routine).

GDI Select exit routine. A program that a user defines to access a non-IBM DBMS that has an SQL interface, or for help performing a two-stage extraction from a DB2 database. GDI select exits let a user submit an SQL-like SELECT statement to DataRefresher without previously storing field descriptions in the FDTLIB. (Contrast with GDI record exit).

General Data Extract feature. This DataRefresher feature (which includes the UIM and the DEM) lets you extract data from an IMS/VS database, or a physical sequential or VSAM file. If you use a generic data interface exit, you can extract data from IBM relational databases without using the REM or from other data sources not directly supported by DataRefresher.

generic data interface (GDI). An interface that accesses MVS databases and files not directly supported by DataRefresher. This source data is accessed via a GDI exit.

generic data interface (GDI) exit routine. A user-written routine that accesses MVS databases, VMS databases, and any files not directly supported by DataRefresher. There are two types of GDI exits: GDI Select exits and GDI Record exits.

generic output interface (GOI) exit routine. A user-written routine that receives extracted data and can be used to convert the data to a user-defined format or written to files not directly supported by DataRefresher.

GOI. generic output interface.

help panel. Information displayed when the user presses the HELP function key while using DataRefresher Dialogs.

hierarchical database. A tree-like, top-down arrangement of segments in a database, beginning at the top of the hierarchy with a root segment and proceeding downward to dependent segments, as in an IMS/VS DL/I database.

high-level language (HLL). A programming language that does not reflect the structure of any particular computer operating system.

High Speed Sequential Retrieval (HSSR). An IMS database tool that delivers efficient access performance to IMS data.

HLL. high-level language.

host. The primary or controlling computer in a multiple computer installation; in this case, the computer running DataRefresher. (Contrast with remote).

HSSR. High Speed Sequential Retrieval.

HUP. Hierarchical Update Program - a DataPropagator NonRelational feature used for propagating from a relational source, such as DB2, to a hierarchical target, such as IMS.
IBM Database 2* (DB2*). A program that provides a full-function relational database management system on MVS and supports access from MVS applications under IMS, CICS*, TSO, or batch environments.

IBM software distribution (ISD). The IBM division that distributes IBM programs.

IMS/VS. Information Management System/Virtual Storage.


INGRES*. A relational database management system that is a product of Relational Technology, Inc.

installation verification procedure (IVP). An IBM program shipped with a product which verifies whether major segments of the product operate correctly following installation of the product.

Integration Exchange Format (IXF). IXF is a self-defining sequential file format providing character and source representation of data that helps applications exchange data.

Interactive System Productivity Facility (ISPF). A program that controls the execution of DataRefresher Dialogs.

internal segment. A repeating group of data within a parent segment; the data can be fixed or variable in length.

ISD. IBM software distribution.

ISPF. Interactive System Productivity Facility.

IVP. installation verification procedure.

IXF. Integration Exchange Format.

JCL. job control language.

JCS. job control statement(s).

JES2. Job Entry Subsystem 2.


Job. A set of computer programs, files, and control statements that are sent to the operating system for processing.

job control language (JCL). A control language used to identify a job to an operating system and to describe the job’s requirements. In DataRefresher, JCL is the control language used to describe required DataRefresher data resources and to run a Data Refresher job. A DataRefresher job stream includes JCL and Data Refresher commands.

job control statement(s) (JCS). JCS is a set of control statements that identify and describe a job to the operating system for routing and final processing of extracted data. JCS can include both JCL and extracted data. The JCS controls the DRU, any load utility, or any application that processes the extracted data.

Job Entry Subsystem 2 (JES2). An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In an installation with multiple processors, each JES2 processor runs independently. See also Job Entry Subsystem 3.

Job Entry Subsystem 3 (JES3). An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In an installation with several loosely coupled processors, JES3 lets the global processor exercise centralized control. See also Job Entry Subsystem 2.

job statement. The job control statement that identifies the beginning of a job. It contains such information as the name of the job, account number, and class and priority assigned to the job.

job step. A unit of work represented by running a single program that resides in the load library. A job can consist of one or more steps.

Julian date. A date format that contains the year in positions 1 and 2, and the day in positions 3 through 5. The day is represented as 1 through 366, right-adjusted, with zeros in any unused high-order positions.

Kanji feature. The DataRefresher program that establishes a Kanji-language environment for DataRefresher Dialogs users.

katakana. A character set of symbols used in one of the Japanese phonetic alphabets. The DataRefresher uppercase feature allows the dialogs to be viewed on terminals which support katakana.

key. (1) One or more characters used to identify the record and establish the order of the record within an indexed file. (2) In VSAM, one or more consecutive characters taken from a data record, used to identify the record and establish its order with respect to other records.

key-sequenced data set (KSDS). A VSAM file whose records are loaded in key sequence and controlled by an index. Records are retrieved and stored by keyed
access or by sequential access, and new records are inserted in key sequence by means of distributed free space.

**keyword.** (1) A part of a DataRefresher command parameter that has a specific meaning to that command and is shown in uppercase letters in the syntax diagram. See also **parameter**.

**KSDS.** key-sequenced data set.

**LE/370 (Language Environment/370).** SAA AD/Cycle Lanaguage Environment/370. A program that allows one high level language to have imbedded calls to routines written in some other high level language.

**load utility.** A program that puts data into one or more tables in a table space or partition.

**Logical Unit (LU).** An interface through which a DataRefresher user accesses the SNA network.

**logical unit name.** The Virtual Telecommunications Access Method (VTAM®) logical unit resource name for initiating communications with the remote node.

**LU.** Logical Unit.

**main storage databases (MSDB).** An IMS/VS Fast Path database used to store and provide access to an installation’s most frequently used data. The data in an MSDB is stored in segments. Each segment can be available to all terminals, or assigned to a specific terminal. To provide fast access and allow frequent update to this data, MSDBs reside in virtual storage during execution. MSDBs cannot be shared.

**Map Capture Communication Area (MCCA).** The control block used for communication between DataRefresher (either the UIM or the DEM) and a user-written map capture exit routine.

**Map Capture exit routine.** A user-written exit routine that can retrieve DataRefresher mapping information for all files and PSBs used during an extract request. This mapping information can be saved for later use, for example, for data propagation.

**master index table (MIT).** A table that contains administrative information for End User Dialogs. For example, the nickname table and the end user table.

**MCCA.** Map Capture Communication Area.

**minimum segment (MINSEGM).** The lowest segment necessary in a hierarchical path to qualify for extraction.

**MINSEGM.** minimum segment.

**MIT.** master index table.

**MSDB.** main storage databases.

**MVG.** Map Verification and Generation - a DataPropagator NonRelational feature used with the DataRefresher UIM when using a DataPropagator NonRelational map capture exit in the DataRefresher SUBMIT command.

**Multiple Virtual Storage (MVS).** An IBM operating system that is in an SAA environment.

**MVS.** Multiple Virtual Storage.

**network job entry (NJE)**. Used in JES2. Allows selected jobs, in-stream (SYSin) data sets, system output (SYSOUT) data sets, operator commands and messages, and job accounting information to be transmitted from one computer system to another.

**network job interface (NJI)**. Used in JES3. Allows selected jobs, in-stream (SYSin) data sets, system output (SYSOUT) data sets, operator commands and messages, and job accounting information to be transmitted from one computer system to another.

**nickname.** In DataRefresher End User Dialogs, a short, convenient name assigned to a specific node or subsystem and the JCL used to route requests to that system.

**NJE.** network job entry.

**NJI.** network job interface.

**node entry.** A name that defines the source and target systems; for example, the node entry in a JCL job control statement or CMSBATCH link.

**null separator field.** The 1-byte field (2 bytes for IXXF output) that begins each data field in an extract output row. DataRefresher puts a hyphen in this field if the data field is null. It is called a separator because it visually separates the data columns in the extract output.

**Online DataRefresher commands.** A set of TSO REXX EXECs that let a user run DataRefresher commands in the TSO foreground; process UIM, DEM, and REM requests.

**ORACLE**. A database management system that is a product of Oracle Corporation.

**packed decimal data type.** A data type in which each byte in the field except the right-most byte represents two numeric digits. The rightmost digit contains one digit and the sign. For example, the decimal value +123 is represented as 0001 0010 0011 1111.
**panel.** A predefined display image. It may be a menu, a data entry panel, or for information only.

**parameter.** A keyword, or variable, or a combination of keywords and variables used with a command to affect its result. In DataRefresher command syntax, required parameters are displayed on the main path of the syntax and optional parameters are displayed below the main path. Default parameters are displayed above the main path of the syntax. See also **keyword.**

**parent segment.** A segment in a database that has one or more dependent segments below it in a hierarchy.

**partial path.** A hierarchical path without an occurrence for every segment.

**partitioned data set (PDS).** A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

**PCB.** Program Communication Block.

**PDS.** Partitioned Data Set.

**persistent extract.** An extract request that is retained in the EXTLIB after an extract runs.

**physical segment.** The smallest unit of accessible data in a database.

**physical sequential file.** A file in which records are processed in the order in which they occur in the file.

**polling interval.** The elapsed time between DEM searches of the EXTLIB. The DEM periodically searches for qualifying extract requests to process. This is used with long-running DEMs.

**Program Communication Block (PCB).** An object that describes a communication block to a program.

**Program Specification Block (PSB).** A set of statements naming the required databases, segments to access, and database modification options for a program. The PSB contains a given program communication block (PCB) for each database named, in the sequence used by the program.

**Program Support Representative (PSR).** An IBM appointed program support technician.

**program temporary fix (PTF).** A temporary solution or bypass of a problem diagnosed by IBM.

**proxy.** An account identifier on a VMS operating system (a product of the Digital Equipment Corporation). The proxy provides IBM users with file access and default privileges on the VAX** computer system.

**PSB.** Program Specification Block.

**PSR.** Program Support Representative.

**PTF.** program temporary fix.

**QMF.** Query Management Facility.

**QSAM.** queued sequential access method.

**Query Management Facility (QMF’).** An interactive query product that lets you create reports and charts from relational data.

**queued sequential access method (QSAM).** A queue containing input data blocks that are awaiting processing or output data blocks that have been processed and are awaiting transfer to either auxiliary storage or to an output device.

**RACF.** Resource Access Control Facility.

**relational database.** A database that is organized and accessed according to relationships between data items. Contrast with **hierarchical database.**

**relational database view.** A relational database view is created by DB2 or SQL/DS and controls access to data on these relational databases.

**Relational Data Extract Feature.** A DataRefresher feature. The REM is used for extracting data from a DB2 or SQL/DS database.

**Relational Extract Manager (REM).** The DataRefresher program that extracts data from a DB2 or SQL/DS database.

**REM.** Relational Extract Manager.

**REM data sources.** Any DB2 or SQL/DS data accessed by the Relational Extract Manager (REM).

**remote.** Pertaining to a system, program, or device that is accessed through a telecommunication line. (Contrast with **host**)

**remote file description table library (RFDTLIB).** A library on a non-IBM computer containing all the DXTFILE and DXTVIEW definitions of data accessible to the non-IBM operating system that are candidates for extracts using DataRefresher. This feature was used by DXT/D1 but is no longer supported.

**Remote Spooling Communications Subsystem (RSCS).** The licensed program that allows the VM system to fully participate in a network of SNA/non-SNA
Network Job Entry (NJE) System nodes, SNA/non-SNA 3270 Information Display System printer nodes, and Bisync Remote Job Entry (RJE) nodes. This capability permits CMS users to transmit and receive spool files or messages to or from any defined node in the network.

repository. An organized group of information that supports business and data processing activities and provides a single point of control for the management and sharing of that information.

Resource Access Control Facility (RACF). An IBM program that provides for controlled access to system resources by identifying and verifying users to the system, authorizing access to DASD data sets, logging detected unauthorized attempts to enter the system, and logging detected accesses to protected data sets.

RFDTLIB. remote file description table library.

root segment. In IMS/VS, the main segment of a database to which all other segments are related. This is the top of the hierarchy tree.

RSCS. Remote Spooling Communications Subsystem.

run mode. Identifies a DEM as either long-running or terminating. The run mode of a long-running DEM determines how long the DEM should run. When in long-running mode the DEM uses a polling interval.

RUP. Relational Utility Program - a DataPropagator NonRelational feature used with the DataRefresher UIM when using a DataPropagator NonRelational map capture exit in the DataRefresher SUBMIT command.

SAA. Systems Application Architecture*.

SAP. Structures Access Program.

session. A long logical connection that allows communication between two logical units using the APPC. (Contrast with conversation)

SFS. shared file system.

SFS Directory. The place where shared file system files are grouped — analogous to a minidisk.

shared file system (SFS). An extension of the CMS file system, that allows simultaneous sharing of CMS programs and data by multiple users and applications.

simple file. A file that contains only one record type. Contrast with structured file.

SMP. System Modification Program.

SQL. Structured Query Language.

SQL/DS. Structured Query Language/Data System.

structured file. A file that contains multiple record types or internal segments or both. (Contrast with simple file).

Structures Access Program (SAP). A DataRefresher program that employs user-specified data structures to generate DataRefresher data description statements, extract request statements, and a statement specifying the creation of a DB2 table to contain the extracted data.

Structured Query Language (SQL*). A language used to communicate with DB2 and SQL/DS.

Structured Query Language/Data System (SQL/DS*). The relational database management system that runs under VM.

synchronous. Occurring with a regular or predictable time relationship.

System Modification Program (SMP). The program used to install DataRefresher under MVS.

Systems Application Architecture* (SAA*). A set of IBM software interfaces, conventions, and protocols that provide a framework for designing and developing applications that are consistent across systems.

table. A named collection of data consisting of a number of named vertical rows and a number of unordered horizontal rows which is under the control of a relational database manager.

time sharing option (TSO). An option on the operating system that provides interactive time sharing from a display station.

transaction. (1) A job or a job step. (2) In IMS, a specific set of input data that starts a specific processor job.

translation table. A table used by DataRefresher that provides replacement characters of one code page for characters of a different code page.

TSO. time sharing option.

TSO/E REXX. The implementation of the Systems Application Architecture (SAA) Procedures Language on the MVS system.

TSO foreground. The environment in which programs are swapped in and out of main storage to let terminal users share processing time.

2-stage extraction. The DataRefresher process for extracting data from a non-relational data source. First, the UIM validates and queues the request in EXTLIB; the DEM then executes the request, depending on the schedule that you have established for the DEM. By
contrast, extract requests from relational data sources are immediately executed by the REM (one-stage).

**UCF.** Uppercase Feature.

**UIC.** user identification code.

**UIM.** User Input Manager.

**Uppercase Feature (UCF).** The DataRefresher program that lets users use the DataRefresher Dialogs on Japanese Katakana terminals (555x, 556x).

**User Data Type exit routine.** A user-written routine that transforms fields in a user-defined format into a data type supported by DataRefresher.

**user identification code (UIC).** A unique identifier for each user on the VMS system.

**User Input Manager (UIM).** The DataRefresher program that validates and enters file and database descriptions into the FDTLIB and validates and enters DEM extrct requests into the EXTLIB.

**value.** Information assigned to a parameter associated with a command or keyword.

**variable.** A part of a DataRefresher command parameter that you supply and is displayed in lowercase letters in the syntax diagram.

**variable-length record.** A record having a length independent of the length of other records with which it is logically or physically associated. (Contrast with fixed-length record).

**view.** See DXTVIEW, DXTVIEW description, or relational database view.

**Virtual Machine (VM).** An IBM system that is part of an SAA environment.

**Virtual Memory System (VMS**").** An operating system produced by Digital Equipment Corporation that is used on the VAX computer.

**Virtual Storage Access Method (VSAM).** An access method for direct or sequential processing of fixed- and variable-length records on direct access devices. A key field (key sequence) can organize the records in a data set or file logically, in the physical sequence in which they are written (entry sequence), or by relative-record number. DataRefresher does not support extraction from a VSAM relative record data set (RRDS).

**VM.** Virtual Machine.

**VMS.** Virtual Memory System.

**VSAM.** Virtual Storage Access Method.
Index

A
accessing a file with your GDI record exit 24
ADMINISTER option in Administrative Dialogs
description 59
administering
   End User Dialogs 161
ead user profile 193
Administrative Dialogs
   ADMINISTER option 59
   CANCEL command 123
   checking status of DEM extract requests 121
deleting data descriptions 115
description 59
   DESCRIPTION option 59
editing
   JCL models 66
   JCS models 66
   end user table (EUT)
      updating 141
   enrolling users
      in MVS 149
      in VM 151
exporting objects 135
EXTRACT option 59
help 161
importing objects 149
JCL/JCS option 59, 66
knowledge required 60
LIST command 124
main menu panel 59
MIT (Master Index Table)
   maintaining 148
   providing access to 148
   REM job to build or update 146
   UIM job to build or update 145
navigating in panels 61
nickname table (NNT)
   building 140
   updating 140
online help 161
overview 50
panel types 61
printing data descriptions 117
PROFILE option 59, 127
punching data descriptions 117
requesting source description data 142
scrolling panels 61
setting processing options 134
setting up End User Dialogs 139
STATUS command (DEM) 121
associating JCL/JCS with an extract request
   priorities for 194
   with Administrative Dialogs 193

B
books
   building an extract request with End User Dialogs
      IMS to DB2 example 165
   joining two DB2 tables into SQL/DS example 177

C
CANCEL command
   canceling extract requests 123
choosing sources
   tables
      joining two DB2 tables into SQL/DS 178
   views
      IMS to DB2 example 166
CLISTs
   D V R E L L G N  63, 163
commands
   End User Dialogs 160
   ISPF 160
CREATE DATATYPE command 16
CREATE DXTFILE command 18
CREATE DXTPSB command 24
CREATE DXTVIEW command 39
creating
   extract request 12

D
DAP (Dictionary Access Program)
   creating data descriptions 107
   creating data descriptions with, 49
   definition 11
   editing output 112
   JCL model DVREDDAP 67
   using from the Administrative Dialogs 110
   using in batch 107
data descriptions
   deleting 115
   printing 117
   punching 117
Data Reformat Utility
   See DRU (Data Reformat Utility)
DataRefresher
   and exit routines 7
   and MVS 4
   and VM 4
   file description
      assigning names 19
DataRefresher (continued)
overview 3
PSB description
assigning names 25
sources 6
sources and targets 4
targets 6
user data type description
source and target data 16
uses of, 3, 9
view description
assigning names 40
assigning types 40
DataRefresher dialogs
accessing DataRefresher dialogs objects 135
building profiles 127
definition 11
directing output in MVS 69
directing output in VM 70
enrolling MVS users 149
enrolling VM users 151
exporting objects 135
extract output options for End User Dialogs 132
help 161
JCL models 66
JCS models 66
object sharing 137
online help 161
processing options for Administrative Dialogs 134
requesting source description data 142
routing output in MVS 70
setting up End User Dialogs 139
specifying default JCL 127
specifying default JCS 128
specifying SUBMIT and EXTRACT options 129
starting a DataRefresher dialogs session 163
starting a session in MVS 163
starting sessions in MVS 63
starting sessions in VM 63
SUBMIT command parameters 133
DataRefresher dialog objects
accessing 135
exporting 135
importing with Administrative Dialogs 149
object sharing 137
SUBMIT command
options for End User Dialogs 131
DataRefresher library
maintaining 138
personal library 138
DB2 and VSAM
joining extracted data with the system editor 55
DB2 database
extracting data with End User Dialogs 177
extracting data with the system editor 53
extracting two DB2 tables into SQL/DS with End User Dialogs 177
DB2 database (continued)
identifying the target table with End User Dialogs 170
JCS model DVREDDBC 68
JCS model DVREDDJC 68
debug levels of the DEM 130
decision support systems
definition 3
DEM (Data Extract Manager)
processing extracts 83
DEM debug levels 129
describing data
correct data 10
DESCRIPTION option in Administrative Dialogs
description 59
diagnostic information 129
Dictionary Access Program
See DAP (Dictionary Access Program)
documentation
DRU (Data Reformat Utility)
DVREDRUM JCS model (MVS) 69
DVREDRUV JCS model (VM) 69
invoking the DRU in MVS 69
invoking the DRU in VM 69
DVREDDAP JCL model
invoking 67
DVREDDBC JCS model
creating or replacing a table 68
invoking the load utility 68
replacing a table 68
DVREDDJC JCS model
invoking the load utility 68
loading data into a table 68
DVREDDXT JCL model
invoking with JCL 67
DVREDEJVM JCS model
invoking the MIT utility 68
DVREDEJLV JCS model
invoking the MIT utility 69
DVREDREM JCL model (MVS)
invoking 67
DVREDREV JCL model (VM)
invoking 67
DVREDRUM JCS model
invoking the DRU 69
DVREDRUV JCS model
invoking the DRU 69
DVREDJJC JCS model
invoking the load utility 68
DVRELLGN CLIST
invoking 63, 163
DVREXUSR EXEC
invoking 63, 163
DXTFILE
creating a description using a model 75
DXTPSB
creating a description using a model  75
DXTVIEW
creating using a model  77

E
End User Dialogs
administering  161
associating
  JCL with an extract request  193
  JCS with an extract request  193
authorization  141
building an extract request (example)  165, 177
choosing source
  DataRefresher views  166
choosing source tables  178
commands  160
description  157
extract output options  132
extracting
  IMS to DB2  165
  extracting from two DB2 tables to SQL/DS  177
function keys  159
identifying the target table (examples)  170, 187
joining two source columns into one target
  column  185
main menu  157
main menu panel
  COLUMNS option  158
  CONDITIONS option  158
  DB ACCESS option  158
  from IMS to DB2  165
  JOIN option  158
  PROFILE REVIEW option  158
  TABLES option  158
  TARGET option  158
navigating through the panels  157
nicknames  161
object sharing  137
overview  50
panels  168
planning an extract  177
Profile Review panel  193
providing database access information  172, 189
saving the extract request  173, 190
security  141
selecting columns  168, 180
sending the extract request  174, 191
setting up with Administrative Dialogs  139
specifying built-in functions  181
specifying conditions  168, 184
specifying SUBMIT command options  131
starting a session (example)  165, 177
end user profile
administering  193
end user profile (continued)
  description  193
end user table (EUT)
  description  139
  updating  141
enrolling users
  in MVS  149
  in VM  151
error message help
  End User Dialogs  161
EXECs
  DVREXUSR EXEC  63, 163
exporting objects with the Administrative Dialogs  135
EXTLIB
  deleting extract request from  123
  listing extract requests  124
EXTRACT
  command options  129
  option in Administrative Dialogs
extract manager
  run  13
extract request
  canceling  121
  checking status  121
  creating using a model  79, 86
  listing  121
  using online commands to create  51
extracting data
  EXTRACT in Administrative Dialogs  59
    from a non-relational source  73
    from a relational source  85
    using the system editor  52
  with End User Dialogs  157
    joining two DB2 tables into SQL/DS  177
  with the SAP  89

F
FDTLIB (file description table library)
  deleting data descriptions  115
  printing data descriptions  117
  punching data descriptions  117
finding information about IMS PSBs and PCBs  27

G
GDI (generic data interface) exits
  editing UIM jobs for end users to access data  145
generic data interface (GDI) exits  167
generic data interface (GDI) exits, example  179

H
help
  End User Dialogs  161
  error message help  161
help (continued)
          panel help  161

I
identifying the target table with End User Dialogs  170
IMS to DB2 example  170
joining two DB2 tables into SQL/DS example  187
import/export dialogs  149
importing objects with Administrative Dialogs  137, 149
IMS DLI database
          extracting data with End User Dialogs  165
          extracting data with the system editor  52
          providing database access information with End User Dialogs  172
IMS to DB2 extract with End User Dialogs  165
informational systems
          definition  3
ISPF commands
          End User Dialogs  160

J
JCL (job control language)
          associating using End User Dialogs  193
          deleting a JCL file  66
          deleting data descriptions  115
          editing models  65
          erasing a JCL file  66
          inline comments  67
          models
            creating new objects using  65
            DVREDDAP  67
            DVREDDXT  67
            DVREDREM (MVS)  67
            DVREDREV (VM)  67
            editing  65
          providing edited files to each user  135
JCL/JCS option in Administrative Dialogs
          deleting a JCL file  66
          description  59
          erasing a JCL file  66
          examples  66
JCS (job control statements)
          associating using End User Dialogs  193
          creating or replacing a table  68
          creating, using a model  78, 85
          editing models  65
          loading data into a table  68
          models
            creating new objects using  65
            DVREDDBC  68
            DVREDDJJC  68
            DVREDEJIM  68
            DVREDEJV  69
            DVREDRUM  69
            DVREDRUM (VM)  69
          JCS (job control statements) (continued)
          models (continued)
            DVREDSJC  68
            editing  65
          job control language
            See JCL (job control language)
          job control statements
            See JCS (job control statements)
          joining two DB2 tables using End User Dialogs  177

L
LIST command
          listing extract requests  124

M
manuals
message help
          End User Dialogs  161
MIT (Master Index Table)
          creating  142
          description  139
          DVREDEJIM JCS model (MVS)  68
          DVREDEJV JCS model (VM)  69
          invoking the MIT utility in VM  69
          maintaining  148
          providing access to  148
          REM job
            for building  146
            for updating  146
          UIM job
            for building  145
            for updating  145
          updating  142
multiple file descriptions  20
MVS (Multiple Virtual System)
          directing output  69
          enrolling DataRefresher dialogs users  149
          invoking the DB2 load utility  68
          invoking the DRU  69
          invoking the MIT utility  68
          invoking the REM  67
          print output  69
          punch output  69
          routing output  70
MVS CLISTs
          DVRELLGN  63, 163

N
navigating in panels
          Administrative Dialogs  61
nickname table (NNT)
          building  140
          description  139
nickname table (NNT) (continued)
  nicknames 161
  updating 140
nicknames
  definition 161
  nickname table (NNT) 161

O
online commands
  creating and running extracts with, 51
operational systems
  definition 3
output
  directing output in MVS 69
  directing output in VM 70
  print output (MVS) 69
  print output (VM) 70
  punch output (MVS) 69
  routing output in MVS 70
  routing output with DataRefresher dialogs 70

P
panels
  display panels 61
  edit panels 61
  navigating 61
  scrolling 61
PCBs, finding information about 27
planning an End User Dialogs extract
  joining two DB2 tables into SQL/DS example 177
print output
  in MVS 69
  in VM 70
profile
  changing 127
  description 59, 127
  panels 127
  setting up 127
PROFILE option in Administrative Dialogs
  changing the profile 127
  extract request output options 132
  processing options for Administrative Dialogs 134
  specifying EXTRACT options 129
  specifying JCL 127
  specifying JCS 128
  specifying SUBMIT command options 131
  specifying SUBMIT options 129
  SUBMIT command parameters 133
Profile Review panel 193
  providing database access information with End User Dialogs
    IMS to DB2 example 172
    joining two DB2 tables into SQL/DS example 189
PSBs, finding information about 27
publications
  punch output
    in MVS 69
    in VM 70

R
Relational Extract Manager
  See REM (Relational Extract Manager)
REM (Relational Extract Manager)
  JCL model DVREDREM 67
  JCL model DVREDREV (VM) 67
  processing an extract 88

S
SAP (Structures Access Program)
  conventions 89
  creating data descriptions 49
  definition 11
  error handling 104
  extracting data with, 89
  output 102
  restrictions
    COBOL 91
    general 90
    IMS 90
    PL/I 90
  return codes 104
  starting 92
saving the extract request with End User Dialogs
  IMS to DB2 example 173
  joining two DB2 tables into SQL/DS 190
scrolling panels
  in Administrative Dialogs 61
selecting columns for an End User Dialogs extract
  IMS to DB2 example 168
  joining two DB2 tables into SQL/DS example 180
sending extract requests with End User Dialogs
  IMS to DB2 example 174
  joining two DB2 tables into SQL/DS example 191
source database 3
  source table description data
    requesting 142
    updating the MIT 142
  specifying conditions with End User Dialogs
    built-in functions 181
    IMS to DB2 example 168
    joining two DB2 tables into SQL/DS example 184
SQL/DS database
  DVREDSJC JCS model 68
  identifying the target table with End User Dialogs 187
starting
  Administrative Dialogs
    in MVS 63

Index 213
starting (continued) Administrative Dialogs (continued)
in VM 63
DataRefresher dialogs  
in MVS 163  
in VM 163  
sessions in VM 163
End User Dialogs
  IMS to DB2 example 165
  in MVS 163  
in VM 163  
  joining two DB2 tables into SQL/DS example 177
STATUS command checking status of extract requests 121
  DataRefresher dialogs  
in MVS 63  
in VM 63
Structures Access Program See SAP (Structures Access Program)
SUBMIT command command options 129 specifying parameters 133 subsystems distinguishing 140 nicknames 140 systems alternate name 140 nickname 140

T
target database 3 types of database informational 3 operational 3 source 3 target 3 types of DXTVIEW descriptions 40

U
UIM (User Input Manager)
  JCL model DVREDDXT 67
User Input Manager
  See UIM (User Input Manager)

V
VM (Virtual Machine)
  directing output 70
  enrolling DataRefresher dialogs users 151
  invoking the DRU 69
  invoking the MIT utility 69
  invoking the SQU/DS load utility 68
  print output 70
Communicating Your Comments to IBM

DataRefresher
Version 1
MVS and VM User's Guide
Publication No. SH19-6996-00

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain only to the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:
  +353 - 1 - 6614246
- If you prefer to send comments electronically, use this network ID:
  - IBM Mail Exchange: IEIBM3FL at IBMMAIL
  - Internet: IEIBM3FL@IBMMAIL.COM

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies.
Readers' Comments — We'd Like to Hear from You

DataRefresher
Version 1
MVS and VM User's Guide
Publication No. SH19-6996-00

Overall, how satisfied are you with the information in this book?

<table>
<thead>
<tr>
<th></th>
<th>Very Satisfied</th>
<th>Satisfied</th>
<th>Neutral</th>
<th>Dissatisfied</th>
<th>Very Dissatisfied</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall satisfaction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How satisfied are you that the information in this book is:

<table>
<thead>
<tr>
<th></th>
<th>Very Satisfied</th>
<th>Satisfied</th>
<th>Neutral</th>
<th>Dissatisfied</th>
<th>Very Dissatisfied</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accurate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Easy to find</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Easy to understand</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Well organized</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applicable to your tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? □ Yes □ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Company or Organization

Phone No.

Address
IBM Software Solutions,
Information Development (IISL),
2 Burlington Road,
Dublin 4,
Ireland.
Program Number: 5696-703

DataRefresher Version 1 Library

SH19-5000  Messages and Codes
LY19-6366  Diagnosis Guide
GH19-6993  An Introduction
SH19-6995  Administration Guide
SH19-6996  MVS and VM User's Guide
SH19-6997  OS/2 User's Guide
SH19-6998  Exit Routines
SH19-6999  Command Reference
GH19-9994  Licensed Program Specifications