Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page vii.

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 (ISSL),
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

<table>
<thead>
<tr>
<th>Notices</th>
<th>vii</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorized use of IBM online books</td>
<td>vii</td>
</tr>
<tr>
<td>Programming Interface information</td>
<td>vii</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>About this book</th>
<th>ix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who should use this book</td>
<td>ix</td>
</tr>
<tr>
<td>Prerequisite information</td>
<td>ix</td>
</tr>
<tr>
<td>How this book is organized</td>
<td>x</td>
</tr>
<tr>
<td>DataRefresher library overview</td>
<td>xi</td>
</tr>
<tr>
<td>Sources of related information</td>
<td>xii</td>
</tr>
</tbody>
</table>

## Part 1. Introducing DataRefresher

<table>
<thead>
<tr>
<th>Chapter 1. DataRefresher overview</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataRefresher and OS/2</td>
<td>4</td>
</tr>
<tr>
<td>DataRefresher and MVS</td>
<td>4</td>
</tr>
<tr>
<td>DataRefresher and VM</td>
<td>4</td>
</tr>
<tr>
<td>Valid DataRefresher data sources and target systems</td>
<td>5</td>
</tr>
<tr>
<td>DataRefresher features</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 2. Preparing for DataRefresher</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Considering your options</td>
<td>13</td>
</tr>
<tr>
<td>Single versus multiple FDTLIBs</td>
<td>16</td>
</tr>
<tr>
<td>Protecting your data</td>
<td>17</td>
</tr>
<tr>
<td>Handling day to day operations</td>
<td>18</td>
</tr>
</tbody>
</table>

## Part 2. Installing DataRefresher on the host

<table>
<thead>
<tr>
<th>Chapter 3. Prerequisite hardware and software</th>
<th>21</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware requirements</td>
<td>21</td>
</tr>
<tr>
<td>Software requirements</td>
<td>22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 4. Installing DataRefresher in MVS</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task overview</td>
<td>25</td>
</tr>
<tr>
<td>Deciding which features to install</td>
<td>26</td>
</tr>
<tr>
<td>Using the installation feature</td>
<td>28</td>
</tr>
<tr>
<td>How System Modification Program Extended (SMP/E) works</td>
<td>29</td>
</tr>
<tr>
<td>Preparing APPC/MVS for communication with DataRefresher OS/2</td>
<td>32</td>
</tr>
<tr>
<td>Migrating from DXT Version 2 Releases 4 and 5</td>
<td>37</td>
</tr>
<tr>
<td>The Migrate Utility</td>
<td>38</td>
</tr>
<tr>
<td>Further installation considerations</td>
<td>45</td>
</tr>
<tr>
<td>DataRefresher installation overviews</td>
<td>46</td>
</tr>
<tr>
<td>After you install DataRefresher</td>
<td>48</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 5. Installing DataRefresher in VM</th>
<th>49</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task overview</td>
<td>49</td>
</tr>
<tr>
<td>Determining which features to install</td>
<td>50</td>
</tr>
<tr>
<td>Allocating the required storage</td>
<td>51</td>
</tr>
<tr>
<td>What you need to know to install the Relational Data Extract feature</td>
<td>52</td>
</tr>
</tbody>
</table>
Chapter 6. Installing the Uppercase and Kanji Features ........................................ 57
National Language Features and the DataRefresher dialogs environment ........... 57
NLF components .................................................................................................. 59
Using double-byte character sets (DBCS) data ............................................... 60
Preparing to install the Uppercase Feature on an MVS system ..................... 60
Preparing to install the Kanji Feature on an MVS system ............................... 62
Preparing to install the Uppercase Feature on a VM system .............................. 63
Preparing to install the Kanji Feature on a VM system ...................................... 64

Part 3. Administering DataRefresher ....................................................................... 67

Chapter 7. General Data Extract feature .............................................................. 71
Task overview ...................................................................................................... 71

Chapter 8. Preparing to use the General Data Extract feature ......................... 75
Extracting data using the General Data Extract feature ............................... 75
Planning to use single or multiple FDTLIBs ................................................... 80
Preparing for Data Extract Manager operation ............................................. 82
Preparing to use exit routines ............................................................................ 84

Chapter 9. Setting up FDTLIB and EXTLIB data sets ........................................ 87
Administering the FDTLIB ............................................................................... 87
Administering the EXTLIB .............................................................................. 92

Chapter 10. Setting up the Data Extract Manager (DEM) ............................... 97
Deciding what type of DEM to run .................................................................. 97
Setting up the DEM environment ................................................................... 98
Determining which extract requests a DEM processes .................................... 102
Information necessary to write DEM JCL and initialization commands ......... 102
Specifying source data sets for DEM extract requests .................................... 103
Specifying output data sets for DEM extract requests .................................... 107
Batching extract requests ............................................................................... 108
Setting up a message table ............................................................................. 111
Administering DEM ....................................................................................... 112

Chapter 11. Protecting your General Data Extract data .................................... 123
Protecting individual data descriptions ............................................................ 123
Controlling access to the FDTLIB and EXTLIB data sets ............................ 129
Preventing unauthorized use of the UIIM and DEM .................................... 130
Data-protection schemes for single or multiple FDTLIBs ............................ 131
Protecting source data .................................................................................... 133
Protecting exit routines .................................................................................. 133
Controlling DataRefresher OS/2 access to host resources ............................ 134

Chapter 12. Administering the Dictionary Access Program ............................... 135
What the Dictionary Access Program does .................................................... 135
What information the DAP needs from the data dictionary ............. 137
**Appendix D. DataRefresher performance considerations** ............... 189
Surveying DataRefresher performance ........................................ 189
Specifying data selection criteria in an extract request ................. 190
Joining data from multiple sources ......................................... 196
Methods to speed extract processing ....................................... 200
Tips for better DataRefresher performance ................................. 202

**Appendix E. DataRefresher dialogs data set and file definitions** ........ 203
MVS common data set definitions ........................................... 203
MVS user-specific data set definitions .................................... 203
VM common file definitions .................................................. 204
VM user-specific file definitions ............................................ 205

Terms and abbreviations ..................................................... 207

Index ..................................................................................... 217

*IBM and OS/2 are trademarks of the International Business Machines Corporation.*
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.

Programming Interface information

This book is intended to help system administrators plan for installation and configuration of DataRefresher.

General-Use programming interfaces allow the customer to write programs that obtain the services of DataRefresher.

General-Use Programming Interface and Associated Guidance Information is identified where it occurs by the following graphic:

______________________ General-Use Programming Interface __________________

General-Use Programming Interface and Associated Guidance Information...

____________________ End of General-Use Programming Interface __________________

Product-Sensitive programming interfaces allow the customer installation to perform tasks such as diagnosing, modifying, monitoring, repairing, tailoring, or tuning of this IBM software product. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM software product. Product-Sensitive programming interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

Product-Sensitive Programming Interface and Associated Guidance Information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

______________________ Product-Sensitive Programming Interface __________________

Product-Sensitive Programming Interface and Associated Guidance Information...

____________________ End of Product-Sensitive Programming Interface __________________
About this book

This book provides planning and administration information for DataRefresher Version 1. It also provides an overview of the procedures for installing DataRefresher on Multiple Virtual Storage (MVS) and IBM* Virtual Machine (VM) operating systems.

The book also includes information about the tasks which have to be performed on a MVS system when you are using DataRefresher under OS/2. For further information about DataRefresher in an OS/2 environment, and how to install DataRefresher in OS/2, see the DataRefresher OS/2 User’s Guide.

Who should use this book

This book is intended for database administrators responsible for planning and administering DataRefresher.

You will find this book helpful if you are responsible for installing or administering any of the following features:

- User Input Manager (UIM)
- Data Extract Manager (DEM)
- Relational Extract Manager
- DataRefresher dialogs (including the Administrative Dialogs and End User Dialogs)
- DataRefresher OS/2
- DataRefresher online commands

Prerequisite information

This book assumes working knowledge of the following:

- Multiple Virtual Storage (MVS)
- IBM Virtual Machine (VM)
- IBM Operating System/2* (OS/2*)
- Time Sharing Option (TSO)
- Conversational Monitor System (CMS)
- Virtual Storage Access Method (VSAM)
- MVS job control language (JCL)
- IBM DATABASE 2* (DB2*)
- Information Management System (IMS)
- SQL/DB System* (SQL/DS*)
- Interactive System Productivity Facility (ISPF)
- Resource Access Control Facility (RACF)
- Communications Manager/2

Notes:

1. The term “MVS” is used in this book to refer to the MVS/ESA environment, except where stated otherwise.
2. The term “VM” is used in this book to refer to the VM/ESA environment, except where stated otherwise.
3. The term “JCL” is used in this book to refer to both job control language in MVS and control statements in VM, except where stated otherwise.
# How this book is organized

This book is organized in parts, each of which covers a specific area of DataRefresher installation and administration. The following table should direct you to the appropriate chapters.

<table>
<thead>
<tr>
<th>For information about</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataRefresher and its features</td>
<td>Part 1, &quot;Introducing DataRefresher&quot; on page 1</td>
</tr>
<tr>
<td>The DataRefresher administrator's role</td>
<td>Chapter 2, &quot;Preparing for DataRefresher&quot; on page 13</td>
</tr>
<tr>
<td>Installing DataRefresher on MVS</td>
<td>Chapter 3, &quot;Prerequisite hardware and software&quot; on page 21 and Chapter 4, &quot;Installing DataRefresher in MVS&quot; on page 25</td>
</tr>
<tr>
<td>Installing DataRefresher on VM</td>
<td>Chapter 3, &quot;Prerequisite hardware and software&quot; on page 21 and Chapter 5, &quot;Installing DataRefresher in VM&quot; on page 49</td>
</tr>
<tr>
<td>Installing DataRefresher on OS/2</td>
<td>DataRefresher OS/2 User's Guide</td>
</tr>
<tr>
<td>Administering the General Data Extract feature</td>
<td>Chapter 7, &quot;General Data Extract feature&quot; on page 71</td>
</tr>
<tr>
<td>Administering DataRefresher OS/2</td>
<td>DataRefresher OS/2 User's Guide</td>
</tr>
<tr>
<td>Administering the Relational Data Extract feature</td>
<td>Chapter 14, &quot;Relational Data Extract feature&quot; on page 145</td>
</tr>
<tr>
<td>Preparing to use DataRefresher online commands and DataRefresher dialogs</td>
<td>Chapter 15, &quot;Setting up online commands&quot; on page 149</td>
</tr>
<tr>
<td>Using DataRefresher in a multilingual environment</td>
<td>Chapter 6, &quot;Installing the Uppercase and Kanji Features&quot; on page 57</td>
</tr>
<tr>
<td>Additional installation and performance considerations</td>
<td>Part 4, &quot;Appendixes, terms and abbreviations&quot; on page 175</td>
</tr>
<tr>
<td>The DataRefresher library</td>
<td>&quot;DataRefresher library overview&quot; on page xi</td>
</tr>
<tr>
<td>Terms used in this book</td>
<td>&quot;Terms and abbreviations&quot; on page 207</td>
</tr>
</tbody>
</table>
# 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>
<tr>
<td><strong>OS/2 User's Guide</strong></td>
<td>SH19-6997-00</td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td><strong>Exit Routines</strong></td>
<td>SH19-6998-00</td>
</tr>
<tr>
<td>This book describes how to write user exit routines to be used by DataRefresher when an extract is processed.</td>
<td></td>
</tr>
<tr>
<td><strong>Command Reference</strong></td>
<td>SH19-6999-00</td>
</tr>
<tr>
<td>This book provides detailed reference information for all of the DataRefresher commands and procedures.</td>
<td></td>
</tr>
<tr>
<td><strong>Messages and Codes</strong></td>
<td>SC19-5000-00</td>
</tr>
<tr>
<td>This book lists the DataRefresher messages with explanations and suggested responses.</td>
<td></td>
</tr>
<tr>
<td><strong>Diagnosis Guide</strong></td>
<td>LY19-6386-00</td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>
Sources of related information

The following books are referenced in this publication:

**IBM Communications Manager/2 publications**

IBM Communications Manager/2 Version 1.0: Configuration Guide. (SC31-6171)
Scenarios. (SC31-6174)
Host Connection Reference. (SC31-6170)
VTAM Resource Definition Reference. (SC31-6412)
VTAM Network Implementation Guide. (SC31-6404)
APPC/MVS Handbook for the OS/2 System Administrator. (GC28-1133)

**IBM Database 2 publications**

IBM DATABASE 2 Version 2: Application Programming and SQL Guide. (SC26-4377)
IBM DATABASE 2 Version 2: SQL Reference. (SC26-4380)
IBM DATABASE 2 Version 2: Database Administration Guide. (SC26-4374)

**ISPF publications**

Interactive System Productivity Facility Dialog Management Guide and Reference. (SC34-4266)

**MVS publications**

MVS/Extended Architecture Integrated Catalog Administration Access Method Services Reference. (GC26-4019)
MVS/ESA Planning: APPC Management. (GC28-1110)
MVS/ESA Planning: Global Resource Serialization. (GC28-1621)
MVS/ESA VSAM Administration Guide. (SC26-4518)

**OS/VS publications**

OS/VS DB/DC Data Dictionary Administration and Customization Guide. (SH20-9174)

**RACF publications**

Resource Access Control Facility (RACF) General Information. (GC28-0722)
Resource Access Control Facility (RACF) Systems Programming Library. (SC28-1343)

**SQL/DS publications**

Database Administration for IBM VM Systems. (GH09-8083)
Database Services Utility for IBM VM Systems. (SH09-8088)

**SMP/E publications**

System Modification Program Extended User’s Guide. (SC28-1302)

**VM/ESA publications**

IBM VM/ESA CP Command and Utility Reference. (SC24-5519)
Part 1. Introducing DataRefresher

This part introduces the DataRefresher product family, and the tasks required to administer DataRefresher.

You will find this part helpful if you need a general introduction to:

- How DataRefresher works
- How to use DataRefresher to optimize data processing at your site

Chapter 1. DataRefresher overview ................................................. 3
DataRefresher and OS/2 ................................................................. 4
DataRefresher and MVS ................................................................. 4
DataRefresher and VM ................................................................. 4
Valid DataRefresher data sources and target systems ................. 5
  Data sources ........................................................................ 5
  Extending the range of data sources ........................................ 5
  Target systems ...................................................................... 7
  Extending the range of targets ............................................... 7
  Manipulating the extracted data .............................................. 7
DataRefresher features ................................................................. 8
  DataRefresher OS/2 ................................................................. 9
  Data Reformat Utility (DRU) .................................................. 9
  DataRefresher Installation feature (MVS only) ....................... 9
  User Input Manager (UIM) ..................................................... 9
  Data Extract Manager (DEM) ................................................. 9
  Migration Utility ................................................................... 10
  Relational Data Extract feature ............................................. 10
  DataRefresher host dialogs ................................................... 10
  National language features .................................................... 10
  Structures Access Program .................................................. 11
  Dictionary Access Program .................................................. 11
  DataRefresher online commands ......................................... 11

Chapter 2. Preparing for DataRefresher ........................................... 13
Considering your options ........................................................... 13
  DataRefresher OS/2 ................................................................. 14
  DataRefresher dialogs ............................................................ 14
  DataRefresher online commands ......................................... 15
  Creating commands with an editor ....................................... 16
Single versus multiple FDTLIBs ............................................... 16
Protecting your data ................................................................. 17
Handling day to day operations ............................................... 18

© Copyright IBM Corp. 1986, 1994
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 or DB2 databases, or in VSAM data sets.

- **Informational (Decision Support) Systems**

  These systems are used by an organization to store the information that can be used to generate reports or perform market analysis. Informational systems are less fixed; they provide the data required for monitoring an organization's business performance. The data can change as the requirements, or emphasis, of the organization changes.

Each system has different sets of requirements and characteristics. 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 system

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

If you use a variety of systems and databases, extracting data with DataRefresher can be easier than writing separate application programs to extract the data. An extract request created with DataRefresher can be submitted or used as a model for creating similar extracts.

DataRefresher provides:

- An object-oriented user interface in the OS/2 environment for defining several data sources and targets, and for creating extracts.
- A variety of methods for defining data sources and creating extract requests in an MVS and a VM environment.
DataRefresher and OS/2

DataRefresher operating in an OS/2 environment provides a Graphical User Interface (GUI) to assist in identifying MVS data sources. From within the GUI you can then choose a target database and create an Extract. An extract is a request to extract data from a source database and place it in a target database.

The GUI is structured to match the skill level and frequency of use with the performance of a task. It reduces the skill level required to extract data from a source and format the data for a target database. The more complex tasks of identifying the sources and targets on the host system are separated from the less technical task of creating the extract. The GUI is the recommended method of creating non-relational data extracts.

The GUI uses a series of Worlds to match the skill level and frequency of use of the task.

For more information, see the DataRefresher OS/2 User’s Guide.

DataRefresher and MVS

DataRefresher operating under MVS provides you with the following features 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 6 provides an overview of the different types of data source that can be used with DataRefresher. It also shows the different types of target systems to which the extracted data can be sent.

**Data sources**

DataRefresher provides you with facilities for extracting data from any of the data sources shown in Figure 1 on page 6:

- 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.
Figure 1. Valid DataRefresher Sources and Targets
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 features

The following is a list of all the features offered by DataRefresher:

- DataRefresher OS/2:
  - Japanese feature

- DataRefresher MVS:
  - Installation feature
  - DataRefresher Base feature:
    - Data Reformat Utility (DRU)
    - DataRefresher online commands
    - DataRefresher dialogs
  - General Data Extract feature:
    - User Input Manager (UIM)
    - Data Extract Manager (DEM)
    - Structures Access Program (SAP)
    - Dictionary Access Program (DAP)
    - Migration Utility
  - Relational Data Extract feature including the Relational Extract Manager (REM)
  - National language features:
    - Uppercase feature
    - Kanji feature

- DataRefresher VM:
  - DataRefresher Base feature:
    - Data Reformat Utility (DRU)
    - DataRefresher dialogs
  - Relational Data Extract feature
  - National language features:
    - Uppercase feature
    - Kanji feature

You do not need to install all of DataRefresher; you need only install those features required by your installation. For example, to create and process a non-relational extract using the GUI, you need only to install DataRefresher OS/2, the DataRefresher base feature, and the General Data Extract feature. See Chapter 2, “Preparing for DataRefresher” on page 13 for information on selecting a configuration.

When creating extracts for a non-relational source, you should use the OS/2 GUI. Use the MVS and VM features of DataRefresher to create relational data extracts, and to process all your extracts. It is possible to create non-relational extracts in an MVS environment, however the GUI has been developed to make the creation of non-relational extracts as efficient and easy as possible, and is therefore recommended.
DataRefresher OS/2
You can use DataRefresher OS/2 to perform many DataRefresher functions on a workstation using the OS/2 end user interface. Data descriptions (called sources and views) and extract requests created using DataRefresher OS/2 can be submitted to a MVS host for processing.

DataRefresher OS/2 communicates with the host system via Logical Unit (LU) 6.2, using the Advanced Program-to-Program Communications (APPC) facilities of OS/2 and MVS. Extensive online help is available to users of DataRefresher OS/2.

To operate DataRefresher OS/2, you must have the General Data Extract feature installed.

Japanese feature
The DataRefresher OS/2 Graphical User Interface will be also be available in Japanese.

Data Reformat Utility (DRU)
The DRU reformats record segments after the data has been sent over the network to another operating system.

Job control language (JCL) requires that inline data extracted from a source is formatted into stream-oriented 80-byte logical records. The DRU reformats the 80-byte record format into a one-row-per-record format.

DataRefresher Installation feature (MVS only)
This feature provides a series of panels where you can enter parameters to install the base product and the other features. Based on the information you enter, the installation feature automatically builds the JCL jobs necessary for installation.

Help is available for each panel interface, should you require assistance.

User Input Manager (UIM)
The UIM validates and stores file and database descriptions in the file description library (FDTLIB). It also validates and puts the Data Extract Manager (DEM) extract requests in the extract request library (EXTLIB). Extract requests stored in the EXTLIB are processed by the DEM.

Data Extract Manager (DEM)
The DEM extracts data from a variety of databases and file types, including:

- IMS databases (MVS)
- VSAM data sets (MVS)
- Physical sequential data sets (MVS)
- Data sources, which are accessed via a GDI exit
Migration Utility

The Migration Utility migrates data descriptions created under Data Extract (DXT) Version 2 Release 5 to DataRefresher Version 1.

Relational Data Extract feature

The Relational Data Extract feature enables you to extract data from DB2 or an SQL/DS database. It contains the Relational Extract Manager (REM), which extracts data from DB2 or SQL/DS databases, running as a DB2 application in MVS or as an SQL/DS application in VM.

DataRefresher host dialogs

DataRefresher provides two types of host dialogs for generating extract requests based on a series of supplied command and JCL models:

End User Dialogs

The End User Dialogs help occasional users, and users with a minimal knowledge of DataRefresher and JCL, to build and submit extract requests. The End User Dialogs administrator provides the users with all the JCL and job control statement (JCS) files and data descriptions required for their extract requests.

Administrative Dialogs

The Administrative Dialogs are intended for use by database administrators or other experienced data processing professionals.

DataRefresher provides models to help you build data descriptions and extract requests. These models contain comments that help you specify the correct commands and values. It also provides models of the JCL and JCS files you can use to store data descriptions and extract requests for DataRefresher. The Administrative Dialogs consist of a series of panels through which you can access these models and edit them to meet your requirements. You can save the extract requests, data descriptions, JCL, or JCS that you create, and use them for future extracts.

The Administrative Dialogs also provide you with the ability to administer all dialog users, in particular the End User Dialogs users.

National language features

The Uppercase feature, and the Kanji feature are two national language features (NLFs) provided with DataRefresher.

To enable you to use these features, DataRefresher lets you enter, process, and display double-byte character set (DBCS) data. The data entered on a DBCS terminal is stored using two bytes for each character. The underlying database management system; IMS, DB2, or SQL/DS (character or graphic data type), provides the actual storage of the 2-byte data.
**Uppercase feature**
With the Uppercase feature you use the DataRefresher dialogs on Japanese Katakana terminals. At these terminals you can type and display Katakana characters; these are Japanese phonetic symbols. You can also type and display uppercase English characters using the Uppercase feature. All fixed text in the dialog panels and messages is translated to uppercase English.

**Kanji feature**
The Kanji feature establishes a Kanji-language environment for the DataRefresher dialog user. In this environment, the DataRefresher dialog panels and user messages are translated into Kanji; a set of commonly used Japanese characters. Kanji characters are part of a double-byte character set (DBCS), where the characters of the set are represented internally by two bytes of storage. Besides Kanji, the Japanese DBCS contains the English alphabet, Hiragana, and Katakana characters.

**Structures Access Program**
The Structures Access Program (SAP) enables the DataRefresher administrator to generate data descriptions and extract requests based on the information that is stored in COBOL, PL/1, or IMS DBD data structures.

You can use SAP to create data descriptions and extract requests for IMS database definitions (DBDs), VSAM files, and physical sequential files.

**Dictionary Access Program**
If your installation uses the IBM OS/VS DB/DC Data Dictionary, you can create DataRefresher File 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.

For other Data Dictionary support, export the data definition to either COBOL or PL/1 format and then use the SAP to generate the data descriptions for DataRefresher.

**DataRefresher online commands**
With DataRefresher online commands, you can:

- Create, delete, and maintain data descriptions
- Print and punch data descriptions
- Build and submit an extract request
- List, cancel, and check the status of submitted extract requests
- Run a single extract request

Instead of submitting the commands as a job for batch execution, you can save time by running the commands online in the TSO foreground. Some DataRefresher online commands let you use an ISPF edit session to edit or review the data required to execute a command. You can also copy DataRefresher models into the edit session to build your data descriptions and extract requests.
Chapter 2. Preparing for DataRefresher

As DataRefresher Administrator you are responsible for controlling and scheduling DataRefresher activities to support your site's operational priorities. You are responsible for providing a smooth working environment for your DataRefresher users, as well as protecting the integrity of your data sources.

This chapter describes your role and responsibilities as DataRefresher Administrator, and describes the options you have for operating DataRefresher. Subsequent chapters in this book describe your administrative tasks in greater detail.

Considering your options

You need to decide what type of DataRefresher configuration best suits your site. DataRefresher offers you the following options:

- DataRefresher OS/2 on a workstation in communicating with an MVS host
- Administrative dialogs in an MVS or VM environment
- End user dialogs in an MVS or VM environment
- Online commands in an MVS environment
- A host editor for creating batch jobs containing DataRefresher commands, in either an MVS or VM environment

You can build and submit extract requests using all, or any combination of, these methods.

You also need to evaluate existing conditions and procedures at your site. This will help you to decide what type of DataRefresher operating environment you need to use. You need to know:

- What level of data-handling experience your users have
- Whether your users will write their own JCL and JCS files
- Whether your users will create their own data descriptions (General Data Extract feature only)
- Whether your users have authority to update databases
- How often your central databases will be updated
- Whether your users will run the same extract requests at regular intervals

If your users know how to use JCL and understand how your source data is organized, consider using one of the following:

- Administrative dialogs
- A non-dialogs environment

If your users have limited system knowledge and restricted access to the source data, consider using one of the following:

- DataRefresher OS/2
- End user dialogs
DataRefresher OS/2

DataRefresher OS/2 offers an easy-to-use workstation interface for performing a range of DataRefresher functions. By manipulating icons representing DataRefresher objects, and filling in information requested by windows, you can register data sources, create data descriptions and build extract requests. These can then be sent to the MVS host for processing.

DataRefresher OS/2 makes DataRefresher functions accessible to a range of users with widely varying knowledge of database operations. Users with little experience of DataRefresher can:

- Queue an extract (that is, a DataRefresher OS/2 extract request) in the Extract Manager
- Store an extract in the EXTLIB (on the host system)

Users with more experience of DataRefresher can also:

- Register a data source (that is a database, data set, or file)
- Create DataRefresher sources and views
- Create an extract using a extract template, or by coping and changing an existing extract

In order to use DataRefresher OS/2 effectively, all users at your site should have some understanding of the following:

- IBM OS/2 products
- The concepts of object-oriented user interface and direct manipulation
- Databases, IMS and SQL
- DataRefresher concepts, such as source, view and extract

You may choose to use DataRefresher OS/2 at your site if you have the General Data Extract feature installed on the host MVS system, and want to offer your DataRefresher users a graphical user interface (GUI) on the workstation for operating DataRefresher.

For information on installing, customizing and operating DataRefresher OS/2, see the DataRefresher OS/2 User's Guide.

DataRefresher dialogs

The DataRefresher dialogs give you the capability to perform a broad spectrum of tasks required to manage data access on an MVS or VM host system. Once they are set up, the dialogs can help to reduce your DataRefresher workload by reducing the amount of input required to create and submit extract requests.

Administrative Dialogs

Working with Administrative Dialogs, you can use the models provided with DataRefresher to create JCL, JCS, and DataRefresher commands. The dialogs will then combine all these commands in a job stream for you.

The following types of models are provided with DataRefresher:

- Data descriptions
- Extract requests
- JCL and JCS
You can tailor the DataRefresher models to suit the data-handling requirements at your site. They contain instructions on how they can be edited to suit your needs.

Although the models are provided with the DataRefresher dialogs, you can browse the models outside of the dialogs (as you would any other data set to which you have access), and you can print the models using your normal procedure for printing a data set. These models can be found in:

MVS environment: A data set called DVR110.DVRJEDIE
VM environment: A macro library called DVRJEDIE MACLIB

**End User Dialogs**
The End User Dialogs help DataRefresher users on the host system by providing prompt panels that generate DataRefresher commands. If you set up the End User Dialogs, your users do not have to code individual jobs and commands to create and submit extract requests.

You might decide to use DataRefresher in a dialogs environment if you:

- Know end users will be creating extract requests on the host system
- Create extract requests on several systems for processing on the host system
- Want assistance submitting your extract requests
- Want to be able to reuse JCL and JCS easily

For information on starting and using the DataRefresher dialogs in the MVS or VM environments, see:

- Chapter 16, “Setting up MVS dialogs” on page 157
- Chapter 17, “Setting up VM dialogs” on page 165
- *DataRefresher MVS and VM User’s Guide*

**DataRefresher online commands**
The DataRefresher online commands enable users on the host MVS system to:

- Create, print, punch, and delete data descriptions
- Build and submit extract requests
- List, check the status of, and cancel submitted extract requests
- Run a single (non-IMS) extract request

The commands are run online in the TSO foreground, rather than being submitted with a job for batch execution. This method saves the user time, and lets the user copy DataRefresher models into an ISPF edit session to build data descriptions and extract requests.

You can change the default parameters used in the DataRefresher online commands so that your users have fewer command entries to type when they operate DataRefresher. Also, command help is provided for the DataRefresher online commands. A user can display the current defaults for a command by issuing the online DataRefresher command and specifying the HELP keyword.
You might decide to use the DataRefresher online commands if your users:
- Are experienced data-processing users
- Run the same extract request at regular intervals
- Use only one system for extract requests
- Do not need help submitting jobs

For information on using the DataRefresher online commands, see:
- Chapter 15, “Setting up online commands” on page 149
- DataRefresher Command Reference

Creating commands with an editor

If you use a system editor to create commands yourself, you are responsible for combining the appropriate JCL, JCS, and DataRefresher commands in a job stream. Extract requests built using this method are jobs submitted for batch execution.

You might decide to use this method if your users:
- Are all experienced data-processing users
- Do not need help submitting jobs

For information on creating DataRefresher commands with an editor, see the DataRefresher MVS and VM User's Guide.

Single versus multiple FDTLIBs

When using the General Data Extract feature to extract data, you must first submit data descriptions of your source data to DataRefresher. These data descriptions describe the format of the file or database from which you are extracting data. After you submit data descriptions to DataRefresher, they are stored in a data set called the file description table library (FDTLIB).

You can use either a single FDTLIB or multiple FDTLIBs at your site. Generally, using a single FDTLIB places responsibility for creating and protecting data descriptions on an administrator, and is designed for maximum control. On the other hand, a scheme for using multiple FDTLIBs can distribute the responsibilities of creating and protecting data descriptions to various user groups.

For more information about using single or multiple FDTLIBs, see “Planning to use single or multiple FDTLIBs” on page 80 and “Data-protection schemes for single or multiple FDTLIBs” on page 131.
Protecting your data

You will need to implement a data-protection scheme to protect your DataRefresher data sources from unauthorized use.

General Data Extract feature data sources
DataRefresher uses the security and auditability features of the MVS and VM operating systems, the IMS database system, and Resource Access Control Facility (RACF®) in an MVS environment to protect your non-relational data from unauthorized use.

Some data sources accessed via a GDI exit routine may have a database management system that supports security functions or provides authorization schemes.

For more information on General Data Extract feature security considerations, see Chapter 11, "Protecting your General Data Extract data" on page 123.

DataRefresher OS/2 data sources
The security scheme in operation on your MVS host system controls access to your data and DataRefresher resources, regardless of whether users are operating DataRefresher on the host or via DataRefresher OS/2. Access to the host machine can be further controlled by Advanced Program-to-Program Communications (APPC) security.

For more information on DataRefresher OS/2 security considerations, see "Controlling DataRefresher OS/2 access to host resources" on page 134.

Relational Data Extract feature data sources
DB2 and SQL/DS relational databases have authorization schemes to protect data, and DataRefresher users extracting data from these databases must have sufficient authority to access the appropriate data.

For more information on Relation Data Extract Feature security considerations, see "Implementing a data-protection scheme for the Relational Data Extract feature" on page 146.

DataRefresher online commands
The DataRefresher online commands use the data-protection facilities that are in effect for DataRefresher data sources.

DataRefresher dialogs
The DataRefresher dialogs use the data-protection facilities that are in effect for DataRefresher data sources.
Handling day to day operations

As DataRefresher Administrator, you will be the focal point for data movement at your site. In general, you will be responsible for:

- Ensuring a smooth working environment for DataRefresher users:
  - Creating and maintaining JCL and JCS data sets and files
  - Creating and maintaining data descriptions (General Data Extract feature only)
  - Creating and maintaining FDTLIB and EXTLIB data sets (General Data Extract feature only)
  - Setting up and administering DataRefresher use at the workstation (if you use DataRefresher OS/2)
  - Setting up and administering the End User Dialogs (if you use the DataRefresher dialogs)

- Protecting the integrity of DataRefresher data sources and targets

You may also be responsible for building, submitting, and maintaining extract requests.

See the following chapters in this book for checklists and more detailed information about your administrative tasks:

- Chapter 7, “General Data Extract feature” on page 71
- Chapter 14, “Relational Data Extract feature” on page 145
- Chapter 15, “Setting up online commands” on page 149
- Chapter 16, “Setting up MVS dialogs” on page 157 and Chapter 17, “Setting up VM dialogs” on page 165
Part 2. Installing DataRefresher on the host

This part provides information you need to install the DataRefresher base product and its features in an MVS or VM environment. For information about installing DataRefresher OS/2 refer to the DataRefresher OS/2 User's Guide. This part also describes hardware and software requirements for installing and using DataRefresher in an MVS, VM, or OS/2 environment.

When you install DataRefresher you need to use the detailed installation procedures described in the Program Directory for DataRefresher.

Chapter 3. Prerequisite hardware and software ........................................ 21
  Hardware requirements ......................................................... 21
    OS/2 hardware requirements ........................................... 21
    MVS and VM hardware requirements ................................. 21
  Software requirements ..................................................... 22
    OS/2 system requirements .................................................. 22
    MVS operating system and support program requirements ........ 22
    VM operating system requirements ................................... 22
    Optional function software ............................................. 22
    Other programs and products .......................................... 24

Chapter 4. Installing DataRefresher in MVS .......................................... 25
  Task overview ..................................................................... 25
  Deciding which features to install ......................................... 26
  Using the installation feature .............................................. 28
  How System Modification Program Extended (SMP/E) works ...... 29
    SMP/E commands ............................................................. 29
    The SMP/E sample environment ....................................... 29
    SMP/E zones and libraries .............................................. 30
    Sharing SMP/E data sets between DataRefresher and Data Extract 30
    Distribution libraries for DataRefresher ......................... 31
    Target libraries for DataRefresher ..................................... 31
  Preparing APPC/MVS for communication with DataRefresher OS/2 32
    Prerequisite hardware and software ............................... 32
    Task overview ............................................................... 33
    Customising APPC/MVS .................................................. 34
    Key Communications parameters ..................................... 37
  Migrating from DXT Version 2 Releases 4 and 5 ................. 37
    The Migrate Utility ....................................................... 38
      Migrating DXT 2.5 data descriptions to DataRefresher ........ 39
      Migrate Utility error conditions .................................. 44
    Further installation considerations ................................. 45
      What you need to know to install the Relational Data Extract feature 45
      Preparing to use Job Entry Subsystem 2 (JES2) .............. 45
      Applying program temporary fixes ............................... 45
      Preparing to re-install DXT ........................................... 45
  DataRefresher installation overviews .................................. 46
    DataRefresher base product task overview ....................... 46
    General Data Extract feature task overview ..................... 47
    DataRefresher OS/2 task overview (host installation) ........ 47
    Relational Data Extract feature task overview ................. 47

© Copyright IBM Corp. 1986, 1994
Chapter 5. Installing DataRefresher in VM

Task overview ........................................ 49
Determining which features to install ............... 50
Allocating the required storage ....................... 51
  Determining storage requirements for the VM machine 51
  Storage requirements for the distribution and production files 51
  Contents of the distribution disk .................. 51
  Contents of the production disk ...................... 52
What you need to know to install the Relational Data Extract feature .......... 52
Migrating from DXT Version 2 Releases 4 and 5 .... 53
Applying program temporary fixes (PTFs) .......... 53
Installing the DataRefresher VM base product and the Relational Data Extract feature .......... 55
  DataRefresher base product task overview .......... 55
  Relational Data Extract feature task overview .... 55
After you install DataRefresher ....................... 56

Chapter 6. Installing the Uppercase and Kanji Features .......... 57

National Language Features and the DataRefresher dialogs environment ... 57
  Information translated from English for an NLF .... 58
  Information not translated from English for an NLF 58
  Information translated for the Uppercase and Kanji Features .......... 58
NLF components ........................................ 59
  How NLF components work ......................... 59
  Naming conventions for NLF components .......... 59
Using double-byte character sets (DBCS) data .... 60
Preparing to install the Uppercase Feature on an MVS system ........ 60
  Distribution and target libraries for the UCF ...... 60
  Task overview ........................................ 61
Preparing to install the Kanji Feature on an MVS system .......... 62
  Distribution and target libraries for the Kanji Feature ...... 62
  Task overview ........................................ 62
Preparing to install the Uppercase Feature on a VM system ........ 63
  Task overview ........................................ 64
Preparing to install the Kanji Feature on a VM system .......... 64
  Production libraries for the Kanji Feature .......... 65
  Task overview ........................................ 65
Chapter 3. Prerequisite hardware and software

This chapter describes the hardware and software required to install and run DataRefresher in an OS/2, MVS, or VM environment. Communication between the OS/2 and MVS environments is made via Advanced Program-to-Program Communication, (APPC); LU 6.2.

Hardware requirements

Depending on the environment you intend to use for running DataRefresher you will have different hardware requirements. These are outlined below:

OS/2 hardware requirements

To run DataRefresher in an OS/2 environment, you need a workstation with sufficient memory to satisfy the combined requirements of OS/2 Version 2 Release 1, Communications Manager/2 Version 1, DB2/2, and any other customer-related applications and storage requirements.

To install DataRefresher you need an LU 2 connection. An LU 6.2 connection is needed to communicate with the MVS machine when DataRefresher is running.

The following list shows the typical installation requirements for DataRefresher in an OS/2 environment:

- An 80486 workstation running OS/2 2.1, with a mouse
- 150MB hard disk
- 16MB of memory

The machine will also require at least 20MB of free disk space.

For information on installing DataRefresher in an OS/2 environment, see the DataRefresher OS/2 User's Guide.

MVS and VM hardware requirements

DataRefresher can be installed and operated on any processor that supports the software systems specified in the section "Software requirements" on page 22.

The processors must have sufficient real storage to satisfy the combined requirements of the data sources, operating system(s), DataRefresher access methods, batch requirements, and other user applications.

The configuration must include enough I/O devices to support the requirements for system output, system residence, and system data sets.

Sufficient direct access storage must be available to satisfy the user information storage requirements. This storage can be any direct access facility supported by the operating system.

The machine will require at least 30MB of disk space with an additional:

- 5MB if the Uppercase feature is used
- 5MB if the Kanji feature is used
Software requirements

This section outlines the software required to install and run DataRefresher and its optional features and functions in an OS/2, MVS, or VM environment. Later releases of software are supported unless otherwise specified.

OS/2 system requirements

The following products are required to install and run DataRefresher in an OS/2 environment:

- OS/2 Version 2 Release 1
- Communications Manager/2 Version 1
- IBM Database 2 OS/2 (5622-044) Version 1

For information on installing DataRefresher in an OS/2 environment, see the DataRefresher OS/2 User's Guide.

MVS operating system and support program requirements

One of the following operating systems is required to install and run DataRefresher:

- MVS/ESA*-JES2 (5695-047) Version 4 Release 2
- MVS/ESA-JES3 (5695-048) Version 4 Release 2

The following products are also required:

- MVS/Data Facility Product (5665-XA3) Version 3 Release 2
- System Modification Program Extended (SMP/E) (5668-949) Version 1 Release 7

VM operating system requirements

The following products are required to install and run DataRefresher in a VM environment:

- VM/ESA (5684-112) Version 1 Release 2
- VM/ESA (5684-112) Version 1 Release 5 370 feature
- Remote Spooling Communications Subsystem (RSCS) (5664-188) Version 3 Release 1

Optional function software

The following tables list the programs required to install and use the optional functions and components of DataRefresher. Later releases are also supported unless stated otherwise.

Base product optional software

The following table lists the software required to run the optional functions included with the DataRefresher base product:

<table>
<thead>
<tr>
<th>Function</th>
<th>Required Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataRefresher Online Commands</td>
<td>• TSO/E (5685-025) Version 2 Release 4</td>
</tr>
<tr>
<td>MVS DataRefresher Dialogs</td>
<td>• Interactive System Productivity Facility/Program Development Facility (ISPF/PDF) (5665-402) Version 3 Release 5</td>
</tr>
</tbody>
</table>
Optional features software

The following tables list the software required to run the optional features of DataRefresher:

**Graphical User Interface (GUI) feature**

<table>
<thead>
<tr>
<th>Function</th>
<th>Required Products</th>
</tr>
</thead>
</table>
| Operating DataRefresher from OS/2 | • Interactive System Productivity Facility (ISPF) (5685-054) Version 3 Release 2  
• Advanced Program-to-Program Communication (APPC) also needs to be installed. APPC is a component of the MVS/ESA operating system. |

**General Data Extract feature**

<table>
<thead>
<tr>
<th>Function</th>
<th>Required Products</th>
</tr>
</thead>
</table>
| Extracting data from IMS/VS DL/I database | • High Speed Sequential Retrieval (HSSR) (5685-093) Version 2 Release 1 (part of the IMS/VS Data Base Tools (DBT) program product), and one of the following:  
  – Information Management System (IMS/ESA) (5685-408) Version 3 Release 1  
  – Information Management System (IMS/ESA) (5685-412) Version 4 Release 1 |
| Extracting data from IMS database operating under CICS* | • CICS/ESA* (5685-083) Version 3 Release 3 |
| Data security | • Resource Access Control Facility (RACF) (5740-XXH) Version 1 Release 9.2 |
| Writing exit routines | • Assembler H (5668-962) Version 2 Release 1  
• OS PL/I Compiler/Runtime Libraries or IBM SAA*/AD CYCLE Language Environment*/370 (5668-910) Version 2 Release 3  
• VS COBOL II (5668-958) Release 3.2 |
| Creating data descriptions with the DAP | • OS/VS DB/DC Data Dictionary (5740-XXF) Version 1 Release 6 |
| Creating data descriptions with the SAP | • OS PL/I Compiler/Runtime Libraries or IBM SAA/AD CYCLE Language Environment/370 (5668-910) Version 2 Release 3 |
Relational Data Extract feature

<table>
<thead>
<tr>
<th>Function</th>
<th>Required Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loading data into or extracting data from DB2</td>
<td>One of the following:</td>
</tr>
<tr>
<td></td>
<td>• IBM DATABASE 2* (5665-DB2) Version 2 Release 3</td>
</tr>
<tr>
<td></td>
<td>• IBM DATABASE 2 (5685-DB2) Version 3</td>
</tr>
</tbody>
</table>

DataRefresher VM feature

<table>
<thead>
<tr>
<th>Function</th>
<th>Required Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using the VM Dialogs</td>
<td>• Interactive System Productivity Facility (ISPF-VM/SP)</td>
</tr>
<tr>
<td></td>
<td>(5684-043) Version 3 Release 2</td>
</tr>
<tr>
<td></td>
<td>• Interactive System Productivity Facility/Program Development Facility (ISPF/PDF-VM/SP) (5684-123) Version 3 Release 2</td>
</tr>
<tr>
<td>Loading data into or extracting data from SQL/DS with the VM Relational Data Extract Feature</td>
<td>One of the following:</td>
</tr>
<tr>
<td></td>
<td>• SQL/Data System (SQL/DS) (5688-103) Program Product Version 3 Release 3 for VM and VSE</td>
</tr>
<tr>
<td></td>
<td>• SQL/Data System (SQL/DS) (5688-103) Program Product Version 3 Release 4 for VM and VSE</td>
</tr>
</tbody>
</table>

Other programs and products

This table lists the functions of other IBM programs that work with DataRefresher, and specifies the product releases required. Later releases are also supported unless otherwise noted.

<table>
<thead>
<tr>
<th>Functions</th>
<th>Required product releases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data propagation between IMS and DB2 databases</td>
<td>• DataPropagator NonRelational MVS/ESA (5696-705) Version 2</td>
</tr>
<tr>
<td>Data propagation between relational databases</td>
<td>• DataPropagator Relational (5622-244) Version 1</td>
</tr>
<tr>
<td>Managing data replication tools</td>
<td>• DataHub (5667-134) Version 1 Release 2</td>
</tr>
<tr>
<td>AS decision support</td>
<td>• Application System (AS) (5648-018) Version 3 Release 2</td>
</tr>
<tr>
<td>QMF report writing</td>
<td>• MVS Query Management Facility (QMF) (5706-254) Version 3 Release 1.1</td>
</tr>
<tr>
<td></td>
<td>• VM Query Management Facility (QMF) (5706-255) Version 3 Release 1.1</td>
</tr>
<tr>
<td>ECF PC-to-Host connection</td>
<td>• MVS Enhanced Connectivity Facility (ECF) (5665-396) Version 1 Release 2</td>
</tr>
<tr>
<td></td>
<td>• VM Enhanced Connectivity Facility (ECF) (5664-327) Version 1 Release 2</td>
</tr>
</tbody>
</table>
Chapter 4. Installing DataRefresher in MVS

This chapter provides information you need to know before you can install the following in the MVS environment:

- The DataRefresher base product
- The General Data Extract feature
- The Relational Data Extract feature

Read this chapter and the DataRefresher program directory before you install DataRefresher.

For information on installing the Uppercase and Kanji Features, see Chapter 6, "Installing the Uppercase and Kanji Features" on page 57.

For information on installing DataRefresher OS/2 see the DataRefresher OS/2 User's Guide.

Task overview

The following overview summarizes the general tasks for installing the DataRefresher and its features on an MVS operating system. For a more detailed list of installation procedures, see "DataRefresher installation overviews" on page 46.

For full installation procedures, see the program directory.

1. Determine which DataRefresher features you want to install.
   
   For information on using the General Data Extract feature, the Relational Data Extract feature, and DataRefresher OS/2, see "Deciding which features to install" on page 26.

2. Ensure that you have the required hardware and licensed programs.
   
   For more information on hardware, software and storage requirements, see Chapter 3, "Prerequisite hardware and software" on page 21.

3. Install DataRefresher using the DataRefresher Installation Feature.
   
   For information on using the DataRefresher Installation Feature, see "Using the installation feature" on page 28.

   For information the functions performed by System Modification Program Extended (SMP/E) during installation, see "How System Modification Program Extended (SMP/E) works" on page 29.

4. Prepare the host system for DataRefresher OS/2 (optional).
   
   For information on setting up the host environment for DataRefresher OS/2, see "Preparing APPC/MVS for communication with DataRefresher OS/2" on page 32.

5. If you are installing the General Data Extract feature, create your FDTLIB and EXTLIB data sets.
   
   For information on creating FDTLIB and EXTLIB data sets, see "Migrating from DXT Version 2 Releases 4 and 5" on page 37.
6. Prepare to use the DataRefresher online commands and DataRefresher dialogs.

   For information, see:
   - Chapter 15, “Setting up online commands” on page 149
   - Chapter 16, “Setting up MVS dialogs” on page 157.

7. Perform administrative tasks for the General Data Extract feature and Relational Data Extract feature.

   For details of the administrative tasks, see:
   - Chapter 7, “General Data Extract feature” on page 71
   - Chapter 14, “Relational Data Extract feature” on page 145

Deciding which features to install

You can install DataRefresher features on an MVS host system at the same time that you install the DataRefresher base product, or at any time after base product installation, if there are no environmental constraints.

The following examples will help you determine whether you need to install the General Data Extract feature, DataRefresher OS/2, and/or the Relational Data Extract feature at your site.

- If you want to extract data from an IMS/VS DL/I database, a VSAM data set or physical sequential data set, or use a Generic Data Interface (GDI) exit routine, you must install the General Data Extract feature on that system. As Figure 2 shows, the General Data Extract feature must reside on the same MVS system as the data to be extracted. In this case, data is being extracted from IMS.

![Diagram](image)

Figure 2. Extracting IMS data with the General Data Extract feature

- If you want to use DataRefresher OS/2 from a workstation to create data descriptions and extract requests to be sent up to the host MVS system, you must install:
  - the DataRefresher base product on the host MVS system
  - the General Data Extract feature on the host MVS system
  - DataRefresher OS/2 on each individual workstation
The relationship between the workstation environment and the host system is shown in Figure 3.

Figure 3. Using DataRefresher OS/2

- If you want to extract data from a DB2 database, you should install the Relational Data Extract feature on that system. The Relational Data Extract feature must reside on the same MVS system as the DB2 database, as shown in Figure 4.

Figure 4. Extracting DB2 data with the Relational Data Extract feature

You can also extract data from a DB2 database by installing the General Data Extract feature and writing a Generic Data Interface (GDI) exit routine.

- If you want to use the IBM OS/VS DB/DC Data Dictionary to create data descriptions, you must install the General Data Extract feature. As shown in Figure 5, the General Data Extract feature must reside on the same MVS system as the Data Dictionary.

Figure 5. Using the IBM OS/VS DB/DC Data Dictionary
If you want to use the Structures Access Program to generate DataRefresher data descriptions using IMS database definitions, COBOL structures, or PL/I structures, you must install the DataRefresher base product and the General Data Extract feature on the same MVS system as the libraries. This is shown in Figure 6:

![Diagram showing MVS, VATAM, SASP command generator, and DataRefresher base product]

*Figure 6. Using the Structures Access Program (SAP)*

**Using the installation feature**

DataRefresher installation is performed by means of the DataRefresher Installation Feature, which is supplied with the DataRefresher base product tape.

The DataRefresher Installation Feature provides a series of panels into which you must enter installation parameters. Based on the information you supply, the tool automatically builds the JCL jobs necessary to install the DataRefresher base product and your DataRefresher features. During installation, the Installation Feature assumes that the installer has the necessary authorization to use the resources concerned.

Some of the information required to install DataRefresher concerns existing resources and products on the host system. For example, the DataRefresher Installation Feature prompts you to enter the unit and volume IDs where you want to unload DataRefresher data sets. The relevant systems programmers and database administrators should be able to provide this information.

The Installation Feature also requests information relating specifically to the new installation of DataRefresher. For example, you will be required to choose names for the datasets to be allocated for DataRefresher, and to make certain decisions about the installation.

Help is available for each installation panel should you require assistance.

To ensure that you have all the information necessary to install DataRefresher and its features on MVS, complete the worksheets in Appendix A, "MVS installation worksheets" on page 177.

For full installation procedures, see the DataRefresher program directory.
How System Modification Program Extended (SMP/E) works

System Modification Program Extended (SMP/E) is the basic tool for installing software changes in MVS. The DataRefresher Installation Feature calls SMP/E as part of the DataRefresher installation process. During installation, the SMP/E controls software changes at the element (module or macro) level by:

- Selecting the proper levels of elements to be installed from a large number of potential changes
- Calling system utility programs to accomplish the installation
- Maintaining records of the installed changes and of the function and service level of every element in the system

This precise control of element levels helps to protect system integrity.

SMP/E installs the elements in two types of libraries:

- **Target libraries**, which contain the executable code that makes up the running system.
- **Distribution libraries**, which contain the master copies of all elements for a system. SMP/E uses the distribution libraries as input to the system process that builds production libraries for a new product. SMP/E also uses them for backup if it is necessary to replace or update part of a running product.

SMP/E commands

After allocating space for your libraries, the DataRefresher installation program uses SMP/E to install DataRefresher as follows:

**RECEIVE command**
Directs SMP/E to read the elements from the IBM product tape, determines which elements are applicable to your system, and then stores those elements in a temporary data set.

**APPLY command**
Directs SMP/E to install the elements, after they have been received, in the appropriate target libraries.

**ACCEPT command**
Approves an element for SMP/E installation in the distribution library. Prior to SMP/E installation in the distribution library, the element is installed in the target library and tested.

DataRefresher supplies the control statements necessary to allocate space for the libraries and run the RECEIVE, APPLY, and ACCEPT commands.

The SMP/E sample environment

DataRefresher provides a sample SMP/E environment, which can be used to install the DataRefresher base product and features for testing. The file DVREJBAL creates the target and distribution libraries for DataRefresher. The same libraries are used for the DataRefresher base product and features. It is recommended that you first install DataRefresher in a unique SMP/E environment that can be discarded after testing is completed.
The sample SMP/E environment creates a consolidated software inventory (CSI) data set called SMPCSI. The SMPCSI data set is a keyed VSAM data set that is divided logically into zones. These zones describe the various subsystems and products in the system.

**SMP/E zones and libraries**

There are three types of zones into which the SMPCSI data set is divided:

- **Global zone**, which describes:
  - The associated target and distribution zones
  - System modifications (SYSMODs) waiting to be installed
  - System utilities that SMP/E uses to install SYSMODs

- **Target zone (DVRT110)**, which:
  - Describes the structure and content of a set of target libraries
  - Names the related distribution zone

- **Distribution zone (DVRD110)**, which:
  - Describes the structure and content of a set of distribution libraries
  - Names the related target zone

Figure 7 shows the relationships between the SMP/E zones and libraries.

![Diagram of SMP/E zones and libraries](image)

Figure 7. SMP/E zones and libraries

If DB2 is installed, you may choose to install DataRefresher in its SMPCSI and zones, or create DataRefresher zones within its SMPCSI.

For more information on SMP/E zones, see *System Modification Program Extended User's Guide*. For information on the DASD space required for the SMP/E data sets, see the DataRefresher program directory.

**Sharing SMP/E data sets between DataRefresher and Data Extract**

The SMP/E jobs provided with DataRefresher allow you to support Data Extract (DXT) along with DataRefresher. DataRefresher Version 1 can be installed in the same SMP/E environment as any release of DXT, without deleting or superceding DXT.

Before running an allocation job you will need to decide whether to use any existing CSIs you may already have set up, or whether to create new ones for DataRefresher Version 1. Instructions for both options can be found in the DataRefresher program directory.
After testing is complete, and DataRefresher Version 1 is successfully installed and in use, you may delete DXT. An example of an SMP/E job to delete DXT Version 2 Releases 4 and 5 is available in the DVR110.DVRSAMPE library.

If you have DXT installed, you should first make a backup copy of your SMP/E data sets if you want to save them. After installation, you can delete the old SMP/E data sets. See the DataRefresher program directory for more information.

**Distribution libraries for DataRefresher**

The distribution libraries for the DataRefresher base product and features are shown in Table 1.

<table>
<thead>
<tr>
<th>Distribution libraries</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVR110.ADVRMACE</td>
<td>Interface parts for exit routines</td>
</tr>
<tr>
<td>DVR110.ADVRMOMS</td>
<td>An individual object module for every DataRefresher module</td>
</tr>
<tr>
<td>DVR110.ADVRDBRM</td>
<td>The database request module (DBRM) used by DataRefresher</td>
</tr>
<tr>
<td>DVR110.ADVRPMSE</td>
<td>ISPF panels, messages, and skeletons for running the DataRefresher dialogs</td>
</tr>
<tr>
<td>DVR110.ADVRPWSE</td>
<td>Workstation files for downloading from the host</td>
</tr>
<tr>
<td>DVR110.ADVRINST</td>
<td>Install tool feature files</td>
</tr>
<tr>
<td>DVR110.ADVRTABL</td>
<td>Install tool feature table files</td>
</tr>
</tbody>
</table>

See the DataRefresher program directory for DASD space requirements for these distribution libraries.

**Target libraries for DataRefresher**

The target libraries for the DataRefresher base product and features are shown in Table 2.

<table>
<thead>
<tr>
<th>Target libraries</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVR110.DVRSAMPE</td>
<td>Data files, control files, and load JCL for sample data, online commands, exits, as well as verification procedures and cataloged procedures used to verify, allocate, and run the DataRefresher product</td>
</tr>
<tr>
<td>DVR110.DVRLOAD</td>
<td>The load modules necessary to run DataRefresher base product and features</td>
</tr>
<tr>
<td>DVR110.DVRDBRM</td>
<td>The database request module (DBRM) used by DataRefresher</td>
</tr>
<tr>
<td>DVR110.DVRIPLIBE</td>
<td>ISPF panels for running the DataRefresher dialogs</td>
</tr>
<tr>
<td>DVR110.DVRMLIBE</td>
<td>ISPF messages for running the DataRefresher dialogs</td>
</tr>
<tr>
<td>DVR110.DVRSLIBE</td>
<td>ISPF skeletons for running the DataRefresher dialogs</td>
</tr>
<tr>
<td>DVR110.DVRJEDJIE</td>
<td>Model extract requests, model data descriptions requests, and model runtime JCL</td>
</tr>
<tr>
<td>DVR110.DVRTADME</td>
<td>The application command table, Master Index Table (MIT) information, and the sample data</td>
</tr>
<tr>
<td>DVR110.DVRTLIBE</td>
<td>Shared objects</td>
</tr>
<tr>
<td>DVR110.DVRINTER</td>
<td>REXX execs and sample exit routines</td>
</tr>
<tr>
<td>DVR110.DVRINTAB</td>
<td>Install feature tables</td>
</tr>
</tbody>
</table>
Table 2 (Page 2 of 2). DataRefresher target libraries

<table>
<thead>
<tr>
<th>Target libraries</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVR110.DVRINCFB</td>
<td>CLIST DS will be copied to DVRINCES</td>
</tr>
<tr>
<td>DVR110.DVRINSKL</td>
<td>Skeleton JCL updated by install tool and written to FTO</td>
</tr>
<tr>
<td>DVR110.DVRINPAN</td>
<td>Panels</td>
</tr>
<tr>
<td>DVR110.DVRINMSG</td>
<td>Messages</td>
</tr>
<tr>
<td>DVR110DVRINFTO</td>
<td>Customised JCL updated by install tool and written to FTO</td>
</tr>
</tbody>
</table>

See the DataRefresher program directory for DASD space requirements for these target libraries.

Preparing APPC/MVS for communication with DataRefresher OS/2

For information on installing DataRefresher OS/2 on an individual workstation see the DataRefresher OS/2 User's Guide.

Communication between the OS/2 and MVS environments for DataRefresher is made via Advanced Program-to-Program Communication; APPC (LU 6.2).

This chapter describes how to prepare APPC/MVS for DataRefresher. The connection scenarios between Hosts and workstations can be many and varied. The following list of communications manuals may help with the configuration work necessary for your situation:

- Scenarios
- Host Connection Reference
- Resource Definition Reference
- VTAM Network Implementation Guide
- APPC/MVS Handbook for the OS/2 System Administrator

Prerequisite hardware and software

This section describes the hardware and software you need on MVS for the APPC communications link.

This list assumes that the hardware and software requirements for Non-SNA 3270 terminals have been satisfied.

Hardware requirements
- None

Software requirements
- MVS/ESA V4.2
- VTAM V3.4.1
- NCP V5.3
Task overview

Use the following checklist to identify the steps you need to follow when installing APPC for DataRefresher on MVS:

1. Customise your VTAM definitions to enable LU6.2 sessions.
2. Add a profile for the DataRefresher TP in APPC/MVS.
3. Take note of key communications parameters, these will then be used in the CM/2 communications configuration on the workstation.

Customising VTAM definitions

Enabling for LU6.2

The following VTAM definition is a sample definition for a single workstation enable for LU6.2

```
******************************************************************************
* SWITCHED MAJOR NODE
* M2TRN031 VBUILD TYPE=SWNET,MAXNO=50,MAXGRP=1
* ******************************************************************************
* IG2R050E PU ADDR=C1, STATION ADDRESS(CAN BE ANY VALUE) C
  IDBLK=05D, ID FOR PC C
  IDNUM=305DE, XID. SEE ABOVE FOR CODING DETAILS C
  DISCNT=NO, HANG-UP ON LU LOGOFF C
  maxDATA=265, C
  MAXOUT=7, C
  SSCP=USSSCS, C
  modemat=SNATERM, DYNAMIC LOGMODE C
  LOGTAB=LOGTAB00, C
  LOGAPPL=IGJZIIL, DIRECT LOGON TO VAMP C
  USSTAB=UT21RL, USSTABLE FOR DUBLIN C
  PUTF=2, ISTATUS=ACTIVE C
  STATOPT=‘SPARE‘ C
  *
* IG2S1052 LU LOCADDR=0,MODETAB=MTCICS, DLOGMOD=APPCPCM
  *
* IG2S1053 LU LOCADDR=2 C
* IG2S1054 LU LOCADDR=3 C
* IG2S1055 LU LOCADDR=4 C
* IG2S1056 LU LOCADDR=5 C
******************************************************************************
```

Figure 8. VTAM Definition for the workstation
The following VTAM definition is a sample definition to define a logmode. The parameters defined here must match those defined in the the CM/2 configuration on the workstation.

```
*LOGON MODE TABLE ENTRY FOR HOST TO PC SESSIONS.

******************************************************************************
* APPCPCLM MODEENT
   LOGMODE=APPCPCLM,
   RUSIZES=X'8989',
   SRCVPAC=X'00',
   SSNDPAC=X'01'
******************************************************************************
```

*Figure 9. VTAM Definition for a logmode*

**Customising APPC/MVS**

**Adding a Transaction Program (TP) profile for DataRefresher**

There are 2 ways to add a TP profile for DataRefresher:

1. The APPC/MVS Administration Utility (ATBSDFMNU)
2. The APPC/MVS Administration Dialog

Both of these methods are documented in chapters 6 and 7 of *Planning: APPC Management*.

The following Administration Dialog screens detail the parameters required for the DataRefresher TP:

```
APP Administration

COMMAND ===>  
Select one of the following with an "S". Then Enter.
Type information. Then Enter.

S  TP Profile Administration
   Current TP Profile
      System file . . SYS1.APCTP

S  Side Information Administration
   Current Side Information
      System file . . SYS1.APCSI

S  Database Token Administration
   Current Database Token
      System file . . SYS1.APCTP

Note: For a list of file names, add an "*" suffix to the partial data set.

F1=HELP   F2=SPLIT   F3=END   F4=RETURN   F5=RFIN   F6=RCHANGE
F7=UP     F8=DOWN    F9=SWAP  F10=REFRESH  F11=RIGHT  F12=CANCEL

*Figure 10. APPC Administration*
Select the TP Profile Administration service to add a new TP profile. DataRefresher does not require the remaining 2 services on this panel (Side Information and Database Token Administration).

![List of TP Profiles](image)

**Figure 11. List of TP Profiles**

Add TP Profile

**Figure 12. Add TP Profile**

**Note:** The TP name you supply here here is also used in the APPC CM/2 configuration in OS/2. The value shown in Figure 12 is an example only.
Add TP Profile

Command ===>

Type information. Then Enter.

Transaction Scheduler: ASCH

To system file . . SYSLAPPCTP________________________

TP Name: DRFTP

Level . . : SYSTEM ID . . . :

Tailor Sysout . . . NO__ Yes/No, perform sysout tailoring
Tailor Account . . . NO__ Yes/No, perform account tailoring
Scheduler Class . . . _____ Scheduled class when attached
                      (enter "** for list of classes)
TP Schedule Type . . STD__ Std (Standard) / Multi
                      (Multitrans)
   Generic userid . . . _____ TP "owner", required for Multitrans

F1=HELP    F2=SPLIT    F3=END    F4=RETURN    F5=RFIND    F6=RCHANGE
F7=UP      F8=DOWN     F9=SWAP   F10=REFRESH F11=RIGHT F12=CANCEL

Figure 13. Add TP Profile

Add JCL for TP Profile

Command ===>

Select one of the following with an "S". Then Enter.

Transaction Scheduler: ASCH

To system file: SYSLAPPCTP

- Edit Create new JCL using edit
- Model Use a JCL model as a base for editing

F1=HELP    F2=SPLIT    F3=END    F4=RETURN    F5=RFIND    F6=RCHANGE
F7=UP      F8=DOWN     F9=SWAP   F10=REFRESH F11=RIGHT F12=CANCEL

Figure 14. Add JCL for TP Profile

Customise and add the following JCL for the TP.

//DRFTP JOB H935112-TSDD, MSGLEVEL=(1,1)
//DRF110 EXEC PGM=DVRS9800
//STPLIB DD DISP=SHR, DSN=DRF110.DVRLOAD
//SYSTSRT DD SYSOUT=Y, FREE=CLOSE
//DXTPRINT DD SYSOUT=Y, FREE=CLOSE
//DXTDUMP DD SYSOUT=Y, FREE=CLOSE
Key Communications parameters

Make a note here of the following key communications parameters; you will need them when you are configuring CM/2 on your workstation.

SNA Network ID

APPC/MVS LU name

DataRefresher TP name

CP/NODE name assigned to workstation

LU name assigned to workstation

Logmode name

Migrating from DXT Version 2 Releases 4 and 5

Many components of DXT Version 2 Releases 4 and 5 are compatible with DataRefresher Version 1. However, to use DXT dialogs data sets and exit routines, you need to do the following:

- Save or rename your existing DXT dialogs user data sets. For information on using DXT dialogs tables and data sets, see "Using tables and data sets from DXT dialogs" on page 157.

- Recompile and link-edit the Assembler exit routine, DVRXAACX. In addition, any Accounting exit routines written to work with DXT must be link-edited to include the new control blocks.

- Recompile and link-edit your Map Capture exit routines to include the new control blocks. For information on compiling and link-editing exit routines, see the DataRefresher Exit Routines.

Migration from DXT to DataRefresher should be performed by a system programmer. For DataRefresher migration jobs, see the DataRefresher program directory.

If your site is planning to extract data from an IMS/VS DL/I database, a VSAM data set, a physical sequential data set, or a data source accessed by a GDI exit routine, you need two permanent data sets:

- File description table library (FDTLIB)
- Extract request library (EXTLIB)

The FDTLIB contains the submitted data descriptions and the EXTLIB contains the submitted extract requests. Neither the FDTLIB or EXTLIB should reside on a DASD shared by multiple processors, unless Global Resource Serialization is used. For information on using Global Resource Serialization, see MVS/ESA Planning: Global Resource Serialization.

The FDTLIB and EXTLIB are created by installation verification procedures. The installation procedures provide the control statements necessary to create both of these libraries. For information on how to define single or multiple FDTLIBs or EXTLIBs after DataRefresher installation, see Chapter 9, "Setting up FDTLIB and EXTLIB data sets" on page 87.
The Migrate Utility

The Migrate Utility runs on the host and on the workstation. The host part uses the DataRefresher PUNCH command to punch out the contents or individual objects from the specified FDTLIB and EXTLIB pair to be migrated. The punch data is then parsed to extract the data to be imported into the DataRefresher System Database. This produces 16 files.

These files contain some of the data to be imported. Other information such as IMS PSB was not held in the DXT libraries. You need to add this information to the files before the import step. This can be done on either the workstation or the Host.

The workstation part of the Migrate Utility transfers the files from the host to the workstation and imports the data into the DataRefresher System Table.

Things to do before running the Migrate Utility
Before running the Migrate Utility you need to complete these steps:

1. Pick an FDTLIB and EXTLIB pair to migrate, or from which to migrate individual objects.

2. Register any IMS Databases, VSAM data sets and Physical Sequential data sets upon which the DXTFILEs and DXTPSBs in the FDTLIB you are migrating are based.

3. Import and run the following queries and forms into Query Manager:

   - DXTWORLD.QRY, DXTWORLD.FRMDXTWORLD.QRY, DXTWORLD.FRMDXTWORLD.QRY, DXTWORLD.FRMDXTWORLD.QRY, DXTWORLD.FRMDXTWORLD.QRY, DXTWORLD.FRM

      This query, when run, will produce a report from which you can choose the target DXT World for the FDTLIB and EXTLIB.

   - USERWRLD.QRY, USERWRLD.FRMUSERWRLD.QRY, USERWRLD.FRMUSERWRLD.QRY, USERWRLD.FRMUSERWRLD.QRY, USERWRLD.FRMUSERWRLD.QRY, USERWRLD.FRM

      This query, when run, will produce a report from which you can choose the target User Defined World for any EXITs referenced by the DXTFILEs, DXTPSBs and Extract Requests in the FDTLIB and EXTLIB.

   - IMSPSB.QRY, IMSPSB.FRMIIMSPSB.QRY, IMSPSB.FRMIMSPSB.QRY, IMSPSB.FRMIMSPSB.QRY, IMSPSB.FRMIMSPSB.QRY, IMSPSB.FRM

      This query, when run, will produce a report from which you can choose the IMS PSB, IMS PCB and PSBLIB which correspond to each DXTPSB being migrated.

   - VSAMPSDS.QRY, VSAMPSDS.FRMVSAMPSDS.QRY, VSAMPSDS.FRMVSAMPSDS.QRY, VSAMPSDS.FRMVSAMPSDS.QRY, VSAMPSDS.FRMVSAMPSDS.QRY, VSAMPSDS.FRM

      This query, when run, will produce a report from which you can choose the VSAM data set or Physical Sequential data set corresponding to each DXTFILE being migrated.
Migrating DXT 2.5 data descriptions to DataRefresher

To migrate your DXT 2.5 data descriptions to DataRefresher complete these steps:

Step 1: Edit the DVRXLMG0 CLIST
The DVRXLMG0 CLIST contains information about what is to be migrated and passes this information to the Migrate Utility.

This CLIST can be found in 'DRF110.DRIVER6.DVRINS.CLIST'. Edit this CLIST with information about what you want to migrate.

Following is the DVRXLMG0 CLIST:

Edit the CLIST and enter the following information:

- **hostname = 'WINMVS3'**
  Replace **WINMVS3** with the name of the host from which you are migrating.

- **fdtlib = 'DXTDEV.SAMPLE.FDTLIB'**
  Replace **DXTDEV.SAMPLE.FDTLIB** with the name of the FDTLIB to be migrated.

- **extlib = 'DXTDEV.SAMPLE.EXTLIB'**
  Replace **DXTDEV.SAMPLE.EXTLIB** with the name of the EXTLIB to be migrated.

- **hlq = 'USERID'**
  Replace **USERID** with the High level Qualifier for the data sets to be created by the Migrate Utility. These data sets will contain the data to be transferred to the workstation and imported into the DataRefresher system database.

- **uimload = 'DRF110.DRIVER6.DVRLOAD'**
  Replace **DRF110.DRIVER6.DVRLOAD** with the DataRefresher Load Library, if your DataRefresher Load Library name is other than the default.

**Note:** When the Migrate Utility will be called from the Install tool, this will not be needed.

- **data_refresher_world = '0000000000'**
  Replace **0000000000** with the DataRefresher World to which the objects are to be migrated.

- **user_defined_world = '0000000000'**
  Replace **0000000000** with the User Defined World to which EXIT objects are to be migrated.

- **Objects to be Migrated**
  You can migrate any of these objects:
  - DXTPSBs
  - DXTFILEs
  - DXTVIEWs
  - DATATYPEs
  - Extract Requests
The CLIST DVRTXLMG0 is set up to migrate the complete contents of the FDTLIB and all the extracts awaiting execution in the EXTLIB. This is specified as follows:

```plaintext
/* Set up objects to be migrated */
dxtspsb = 'DXTPSB=*'
dxtfile = 'DXTFILE=*'
dxtview = 'DXTVIEW=*'
datatype = 'DATATYPE='
extid = 'EXTID='
```

Objects can be excluded from migration by removing the asterisk in the statement. For example, the following will migrate DXTFILE only:

```plaintext
/* Set up objects to be migrated */
dxtspsb = 'DXTPSB='
dxtfile = 'DXTFILE=*'
dxtview = 'DXTVIEW='
datatype = 'DATATYPE='
extid = 'EXTID='
```

Specific objects can be migrated by specifying the name of the object. For example, the following will migrate all DXTFILEs and the DXTVIEW VIEW1:

```plaintext
/* Set up objects to be migrated */
dxtspsb = 'DXTPSB='
dxtfile = 'DXTFILE=*'
dxtview = 'DXTVIEW=VIEW1'
datatype = 'DATATYPE='
extid = 'EXTID='
```
Step 2: Running the Migrate Utility
To run the Migrate Utility, you can either enter the following on the ISPF command line:

```
TSO EX 'DRF110.DRIVER6.DVRINS.CLIST(DVRXLMG0)'
```

or enter the following from the TSO Command Processor, Option 6 on the ISPF Primary Options Menu:

```
EX 'DRF110.DRIVER6.DVRINS.CLIST(DVRXLMG0)'
```

where `DRF110.DRIVER6.DVRINS.CLIST` is the name of your CLIST library if it differs from the default.

The Migrate Utility will display progress messages to indicate its status.

On completing successfully, the Migrate Utility will have produced the following data sets:

- hlq.PC.DATAEXIT
- hlq.PC.DATATYPE
- hlq.PC.DXTFILE
- hlq.PC.DXTPCB
- hlq.PC.DXTPSB
- hlq.PC.ERLINK
- hlq.PC.ERVIEWS
- hlq.PC.EXIT
- hlq.PC.EXITDD
- hlq.PC.EXITFILE
- hlq.PC.EXITPCB
- hlq.PC.EXTRACT
- hlq.PC.GDIEXIT
- hlq.PC.MAPEXIT
- hlq.PC.VIEWF
- hlq.PC.VIEWP

where `hlq` is the value entered in DVRXLMG0.

Step 3: Add Required Data to the IMPORT Files
It is not possible to get all the information required to complete the tables from the FDTLIB and EXTLIB alone. Information such as the IMSPSB, for example, has never been held in the DXT libraries. This information is necessary to connect the DXTPSBs and DXTFILEs with the IMS databases and VSAM/PSDS data sets. This information must be provided before the data can be imported into the DataRefresher System Database, because of the referential integrity constraints that exist.

Following is a table which shows the files which require this additional data, the data to be provided, and the column positions in each data set. Edit the data sets and add the required information. (See “Things to do before running the Migrate Utility” on page 38.)
<table>
<thead>
<tr>
<th>Filename</th>
<th>Data required</th>
<th>From column</th>
<th>To column</th>
</tr>
</thead>
<tbody>
<tr>
<td>hlq.PC.DXTPCB</td>
<td>IMSPSB</td>
<td>73</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>IMSPCB</td>
<td>82</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>PSB_LIB_NAME</td>
<td>88</td>
<td>131</td>
</tr>
<tr>
<td>hlq.PC.DXTFILE</td>
<td>VSAM_PSDS_DB</td>
<td>88</td>
<td>97</td>
</tr>
<tr>
<td>hlq.PC.EXIT</td>
<td>LOAD_LIB</td>
<td>19</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td>EXIT_LANG</td>
<td>66</td>
<td>66</td>
</tr>
<tr>
<td>hlq.PC.GDIEEXIT</td>
<td>DB2_SUBSYSTEM</td>
<td>21</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>DB2_PLAN</td>
<td>26</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>UIM_PARMS</td>
<td>35</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>DEM_PARMS</td>
<td>101</td>
<td>194</td>
</tr>
<tr>
<td>hlq.PC.EXITDD</td>
<td>DD_NAME</td>
<td>19</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>DSN_NAME</td>
<td>28</td>
<td>71</td>
</tr>
</tbody>
</table>

**Step 4: TRANSFER the files to the workstation**

A command file RECDB.CMD is provided in X:\DRIVER6\MIGRATE which when run on the workstation will download the data sets produced by the Migrate Utility.

To transfer the files to the workstation follow these steps:

1. Switch to the X:\DRIVER6\MIGRATE directory on the workstation.

2. On your MVS session, go to the TSO Command Processor Option. Select option 6 from the ISPF Primary Options Menu to display the TSO command processor panel.

3. Run the RECDB.CMD command file to transfer the files from your MVS session to the workstation.

The RECDB.CMD command file has 3 parameters which are optional, they are:

- **/D=dir**
  
The directory to where the import files will be transferred. The DEFAULT is D:\MIGRATE.

- **/A=qualifier**
  
The high level qualifier of the files to be transferred. The DEFAULT is DVR110.

- **/S=session**
  
The host emulation session to be used for the file transfer. The DEFAULT is A.

**Note:** You may prefer to edit the files on the workstation; if this is the case, perform the Transfer step followed by the Edit step.
Step 5: IMPORT the files
The order in which the files are imported is important because of the referential integrity constraints that exist between the tables.

Before the files are imported, the IMS databases and VSAM and PSDS data sets on which the DXTPSBs and DXTFILEs are based must be registered in the relevant WORLDs.

A command file IMPORT.CMD is provided in X:\DRIVER6\MIGRATE to import the files in the correct order into the DataRefresher System Database.

To invoke the IMPORT command follow these steps:
1. Switch to the X:\DRIVER6\MIGRATE directory on the workstation.
2. Run the IMPORT.CMD command file to import the data.

The IMPORT.CMD command file has 4 parameters which are optional, they are:

- /D=dir
  The directory from where the data will be imported. The DEFAULT is D:\MIGRATE.

- /A=qualifier
  The qualifier used for the tables in the database. The DEFAULT is DVR110.

- /N=name
  The name of the database. The DEFAULT is DVRSYSDB.

- /M=messagefile
  The name of the file to which the IMPORT Utility sends messages The DEFAULT is IMPORT.MSG.

The IMPORT.CMD command file uses a file called IMPORT.DOC. This file contains information about the data to be imported, the tables and the column positions. This file should not be changed and should reside in the same directory as the command file.

The IMPORT.CMD command file calls the DB2/2 IMPORT API. The messages returned by the API for each file imported are concatenated into the message file specified. The default is IMPORT.MSG.
Migrate Utility error conditions

Migrate Utility error conditions can occur at either the host or the workstation, and could result in the migration not completing successfully.

Host error conditions

If the UIM finishes with a return code of 8 or higher at the host, the Migrate Utility will stop processing. The temporary data sets allocated for running the UIM will not be deleted. These are called:

- hlq.MIGRATE.DXTIN
- hlq.MIGRATE.DXTPRINT
- hlq.MIGRATE.DXTDUMP
- hlq.MIGRATE.DXTPUNCH

The Migrate Utility will also stop processing if an error is detected while allocating, reading, writing or deallocating files.

The messages in the DXTPRINT data set will indicate why the UIM did not complete successfully. Use these messages to diagnose the problem, correct the error condition and then retry the procedure.

Workstation error conditions

The Migrate Utility on the workstation concatenates messages from the DB2/2 IMPORT API into one file (default is IMPORT.MSG). Messages can be produced during import for a number of reasons:

- If duplicates are found
- If the data added to the import files does not match the registered IMS Databases or VSAM/PSDS data sets
- If any views or extract requests which are being migrated are obsolete. That is, the DXTFILEs or DXTPSBs/DXTPCBs (views), or DXTVIEWs (extracts) on which they are based, do not exist.

Use the error messages to diagnose the problem, correct the error condition and then retry the procedure.
Further installation considerations

This section contains information on the following:

- DB2 knowledge required to install the Relational Data Extract feature
- Using Job Entry Subsystem 2 (JES2)
- Applying program temporary fixes (PTFs)
- Re-installing DXT

What you need to know to install the Relational Data Extract feature

Although DataRefresher was designed so that it could be installed by users with a minimum of DB2 knowledge, you do need some knowledge of DB2 when you install the Relational Data Extract feature.

You need to know about:

- Subsystem ID or name. For information on subsystem IDs or names, see IBM DATABASE 2 Version 2: Database Administration Guide.

- The meaning of application plan, DBRM, and bind. See IBM DATABASE 2 Version 2: Application Programming and SQL Guide.

- DB2 security mechanism. You should understand what SYSADM authority is and how to grant and revoke authority. You should also understand the meaning of granting authority to PUBLIC. For information on this, see IBM DATABASE 2 Version 2: Database Administration Guide.

- GRANT and RUN statements. These statements are used in the DataRefresher installation procedure. For information on the GRANT and RUN statements, see IBM DATABASE 2 Version 2: SQL Reference.

Preparing to use Job Entry Subsystem 2 (JES2)

If you are using IBM's Job Entry Subsystem 2 (JES2), you may want to define more internal readers so that the DEM can process more extract requests at a time (batching). JES3 users do not need to do this.

Applying program temporary fixes

When applying a program temporary fix (PTF) to your system, you should read the PTF cover letter to determine which modules are being changed. If you are changing a module that went through the bind process during installation of DataRefresher, then you must rebind that module.

Preparing to re-install DXT

Before you delete or overwrite any previous release of DXT, make a backup copy of the FDTLIB for that release. If you later decide to re-install DXT, use the copy of the old FDTLIB, and follow the installation procedures given in the appropriate DXT program directory.
**DataRefresher installation overviews**

The following overviews summarize the tasks for installing the DataRefresher base product, the General Data Extract feature, the Relational Data Extract feature, and DataRefresher OS/2 on the MVS host system. You can install the DataRefresher features at the same time that you install the DataRefresher base product. To help you gather the information necessary for installation, use the worksheets provided in Appendix A.

For information on installing the Uppercase and Kanji Features, see Chapter 6, “Installing the Uppercase and Kanji Features” on page 57.

**DataRefresher base product task overview**

The following is a task overview of the installation process for the DataRefresher MVS/ESA base product. You can use the space at the left of each step to check it off when the step is completed.

1. Read the DataRefresher program directory
2. Complete the installation worksheets in Appendix A for the DataRefresher base product and the features you are installing
3. Retrieve files from the ISD tape
4. Allocate space for target and distribution libraries for the DataRefresher Installation Feature
5. Run SMP/E RECEIVE and APPLY for the DataRefresher Installation Feature
6. Run the DataRefresher Installation Feature
7. Allocate space for target and distribution libraries for DataRefresher
8. Run SMP/E RECEIVE and APPLY for DataRefresher
9. Make DataRefresher dialogs an ISPF/PDF application to install the base product
10. Modify and copy CLISTs from DVRSAMPE to CLIST library
11. Enroll installer to be able to run the installation verification procedure (IVP)
12. Run job to initialize DataRefresher dialogs environment (optional)
13. Install sample data
14. Run the IVPs for the base functions
15. Run job to clean up DataRefresher dialogs environment (optional)
16. If migrating from DXT, run the dialogs Master Index Table Migration Utility
17. Run SMP/E ACCEPT for the DataRefresher base product

For full installation procedures, see the DataRefresher program directory.
General Data Extract feature task overview

During DataRefresher base product installation, you can also install the General Data Extract feature. The following are the additional steps you must take to install the General Data Extract feature. You can use the space at the left of each step to check it off when the step is completed.

- 1. Run SMP/E RECEIVE and APPLY for the General Data Extract feature
- 2. Run special link-edit jobs for SAP, DAP, and HSSR support programs
- 3. Make JES2 and RACF changes
- 4. Install sample data
- 5. Run the General Data Extract feature IVPs
- 6. If migrating from DXT, recompile and link-edit exit routines
- 7. Run SMP/E ACCEPT for the General Data Extract feature

For full installation procedures, see the DataRefresher program directory.

DataRefresher OS/2 task overview (host installation)

During DataRefresher base product installation, you can also prepare the host system for communication with DataRefresher OS/2. The following are the additional steps you must perform on the host to install DataRefresher OS/2. You can use the space at the left of each step to check it off when the step is completed.

- 1. Run SMP/E RECEIVE and APPLY
- 2. Set up VTAM and APPC/MVS for APPC
- 3. Define the appropriate LU and logon mode to VTAM and APPC
- 4. Define a transaction program
- 5. Establish a batch job class for DataRefresher (optional)
- 6. Perform RACF administration tasks

For full installation procedures, see the DataRefresher program directory.

Relational Data Extract feature task overview

During DataRefresher base product installation, you can also install the Relational Data Extract feature. The following are the additional steps you must take to install the Relational Data Extract feature. You can use the space at the left of each step to check it off when the step is completed.

- 1. Run SMP/E RECEIVE and APPLY for the Relational Data Extract feature
- 2. Modify and run job to bind the Relational Data Extract feature to DB2
- 3. Install sample data
- 4. Modify and run Relational Extract Manager IVP
- 5. Run SMP/E ACCEPT for the Relational Data Extract feature

For full installation procedures, see the DataRefresher program directory.
After you install DataRefresher

The tasks you must perform to administer DataRefresher at your site will vary according to the DataRefresher configuration you have chosen. Besides administering the General Data Extract feature and/or the Relational Data Extract feature, you might also need to administer the DataRefresher online commands and start the DataRefresher dialogs.

For details of administrative tasks, see:

- Chapter 7, “General Data Extract feature” on page 71
- Chapter 14, “Relational Data Extract feature” on page 145
- Chapter 15, “Setting up online commands” on page 149
- Chapter 16, “Setting up MVS dialogs” on page 157
Chapter 5. Installing DataRefresher in VM

This chapter provides the information you need to know before you can install the DataRefresher base product and the Relational Data Extract feature in either of the supported VM environments - VM/ESA or VMSES/E. Read this chapter and the DataRefresher program directory before you install DataRefresher.

For information on installing the Uppercase and Kanji Features, see Chapter 6, "Installing the Uppercase and Kanji Features" on page 57.

Task overview

The following overview summarizes the general tasks for for installing DataRefresher on a VM operating system. For a more detailed list of installation procedures, see either "Installing the DataRefresher VM base product and the Relational Data Extract feature" on page 55.

For full installation procedures, see the program directory.

1. **Determine which features you want to install.**
   
   For information on the functions performed by the DataRefresher base product and the Relational Data Extract feature, see "Determining which features to install" on page 50.

2. **Ensure that you have the required hardware and licensed programs.**
   
   For information on hardware, software and storage requirements, see Chapter 3, "Prerequisite hardware and software" on page 21.

3. **Allocate the required storage.**
   
   For information on DataRefresher storage requirements and the contents of the distribution and production disks, see "Allocating the required storage" on page 51.

4. **Install DataRefresher.**
   
   For information on what you need to know about SQL/DS to install the Relational Data Extract feature, see "What you need to know to install the Relational Data Extract feature" on page 52.

5. **Apply any necessary program temporary fixes (PTFs).**
   
   For information on applying PTFs, see "Applying program temporary fixes (PTFs)" on page 53.

6. **Start the DataRefresher dialogs (optional).**
   
   For information on starting the DataRefresher dialogs, see Chapter 17, "Setting up VM dialogs" on page 165.
Determining which features to install

This section discusses the DataRefresher programs you need to install before DataRefresher can extract data. One of the supported VM base products, VM/ESA or VMSES/E, must be installed before you can install a DataRefresher feature. The DataRefresher base product and any DataRefresher features must be installed on the same system as the source data.

The following examples will help you determine how to install DataRefresher to let users at your site extract data.

- If you want to extract data from a Structured Query Language/Data System (SQL/DS) database under VM, you must install the Relational Data Extract feature on that VM system. As shown in Figure 15, the Relational Data Extract feature must reside on the same system as the SQL/DS database.

![Diagram of VM with SQL/DS and DataRefresher features](image)

*Figure 15. Extracting SQL/DS data*

- If you want to send data across the network, to a CMS file on a VM system, or a physical sequential data set on an MVS system, then you may need the Data Reformat Utility (DRU) to recombine the 80-character record segments. The DRU is part of the DataRefresher MVS/ESA and VM/ESA base products. The DataRefresher base product must be installed on the target system, as shown in Figure 16.

![Diagram of MVS or VM with DataRefresher features](image)

*Figure 16. Sending data to another system*
Allocating the required storage

Before installing DataRefresher in VM, you must allocate storage for:

- The VM machine
- Distribution and production files
- DataRefresher dialogs ISPF Dialog Manager MACLIBs

The DataRefresher program directory provides full storage estimates.

Determining storage requirements for the VM machine

The DataRefresher base product, the Relational Data Extract feature, and the Uppercase Feature require at least 5MB of storage on the VM system. This estimate includes the storage required to run ISPF/PDF for the DataRefresher dialogs and the Relational Data Extract feature.

Storage requirements for the distribution and production files

Storage must be allocated for the DataRefresher distribution and production files on minidisk or in the shared file system. In either case, both types of storage must be disk storage (DASD). The production disk contains online files needed to run the DataRefresher dialogs. If the Relational Data Extract feature is installed, the production disk also contains online files necessary to run this feature. In addition, the production disk contains EXECs for running Installation Verification Procedures (IVPs) and invoking the DataRefresher dialogs (these may be modified for your environment).

The distribution disk contains identical backup copies of everything put on the production disk except for MACLIBs and text libraries (TXTLIBs). In addition, each individual TEXT deck is put on the distribution disk. These TEXT decks are used when applying PTFs.

The estimated DASD requirements are contained in the DataRefresher program directory.

Note: You must allocate the same amount of storage whether or not you install the Relational Data Extract feature.

Contents of the distribution disk

The distribution disk contains files for installation (including installation verification) and maintenance, as shown in Table 3.

<table>
<thead>
<tr>
<th>Table 3. Contents of the distribution disk</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Distribution Libraries</strong></td>
</tr>
<tr>
<td>DVRxxxxx TEXT</td>
</tr>
<tr>
<td>DVRxxxxx yyyy</td>
</tr>
</tbody>
</table>
Contents of the production disk

The production disk contains DataRefresher runtime libraries and files. The libraries and files that the production disk contains after installation are shown in Table 4.

<table>
<thead>
<tr>
<th>Production Libraries</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVRExxx EXEC</td>
<td>Installation procedures, where xxx represents the installation procedures</td>
</tr>
<tr>
<td>DVRIO000 LOADLIB</td>
<td>Code to handle DataRefresher I/O operations</td>
</tr>
<tr>
<td>DVRDATAE MACLIB</td>
<td>DataRefresher sample data descriptions</td>
</tr>
<tr>
<td>DVRJEDIE MACLIB</td>
<td>Model extract requests, data description requests, JCL/JCS and SAP skeletons</td>
</tr>
<tr>
<td>DVRMLIBE MACLIB</td>
<td>ISPF messages for running DataRefresher dialogs</td>
</tr>
<tr>
<td>DVRPLIBE MACLIB</td>
<td>ISPF panels for running DataRefresher dialogs</td>
</tr>
<tr>
<td>DVRSLIBE MACLIB</td>
<td>ISPF skeletons for running DataRefresher dialogs</td>
</tr>
<tr>
<td>DVRADME MACLIB</td>
<td>ISPF table for commands</td>
</tr>
<tr>
<td>DVRTLIBE MACLIB</td>
<td>Table library for shared objects library</td>
</tr>
<tr>
<td>DVRE0000 MODULE</td>
<td>If the Relational Data Extract feature is installed, contains the REM</td>
</tr>
<tr>
<td>DVRRI0000 MODULE</td>
<td>The Data Reformat Utility (DRU)</td>
</tr>
<tr>
<td>DVRIO000 TXTLIB</td>
<td>The text modules required to build the DVRIO000 LOADLIB</td>
</tr>
<tr>
<td>DVRLOAD TXTLIB</td>
<td>The text modules required to run DataRefresher dialogs</td>
</tr>
<tr>
<td>DVRQ0000 TXTLIB</td>
<td>The text modules required to update the master index table</td>
</tr>
<tr>
<td>DVRRT0000 TXTLIB</td>
<td>The text modules required to migrate the master index table</td>
</tr>
</tbody>
</table>

What you need to know to install the Relational Data Extract feature

To install the Relational Data Extract feature, you need some knowledge of SQL/DS:

- Identifying programs and user IDs to SQL/DS through the CONNECT command. You should understand why the CONNECT command is issued and what it does. See SQL/Data System Database Administration for IBM VM Systems.

- Understanding DBA authority. See SQL/Data System Database Administration for IBM VM Systems.

- Preprocessing a program. All application programs that contain SQL commands must be preprocessed. Information about preprocessing a program is in SQL/Data System Database Administration for IBM VM Systems.

- The SQLDSU utility. See SQL/Data System Database Services Utility for IBM VM Systems.

You must have SQLDBA authority to preprocess the REM when you install the DataRefresher Relational Data Extract feature. For information on SQLDBA authority, see SQL/Data System Database Administration for IBM VM Systems.
Migrating from DXT Version 2 Releases 4 and 5

Many components of DXT Version 2 Releases 4 and 5 are compatible with DataRefresher Version 1. However, if you have DXT installed at your site, you should do the following:

- Save your DXT dialogs user MACLIBs. For information on using DXT dialogs tables and MACLIBs, see "Using tables and MACLIBs from DXT dialogs" on page 165.
- Allocate new production and distribution minidisks on which to install DataRefresher Version 1.

For DataRefresher Version 1 migration jobs, see the DataRefresher program directory.

Applying program temporary fixes (PTFs)

DataRefresher is serviced by emergency program temporary fixes (PTFs).

If you have to apply an emergency PTF to your system, you must have the following disks available:

- DataRefresher base product DIST disk accessed in write mode as the 'N' disk
- DataRefresher base product PROD disk accessed in write mode as the 'A' disk
- DataRefresher UCF PROD disk accessed in write mode as the 'M' disk (required only if the UCF is installed)
- DataRefresher Kanji Feature PROD disk accessed in write mode as the 'K' disk (required only if the Kanji Feature is installed)

Use the DVREXBSR EXEC on the production disk to install the PTF. To call DVREXBSR, type DVREXBSR at the CMS prompt.

The EXEC prompts you for the filemode of a temporary disk. For information about obtaining temporary disk space, see *IBM Virtual Machine/Enterprise Systems Architecture CP Command and Utility Reference*.

Most emergency service will require only one EXEC, DVREXBSR. However, if the Relational Data Extract feature is installed, three other EXECs may be required. They are:

**DVREXRPR** Runs the SQL DBSU either to bind DVRXSEDEV ASMSQL to the SQL/DS database, or to reload the DVRXCEDV COPY program to the SQL/DS database. The function performed depends upon the level of SQL/DS installed. To run this EXEC, five input parameters are required. Use the worksheet in Appendix B, "VM installation worksheet" on page 183 to help you gather these parameters.
The parameters are positional and are as follows:

1. DataRefresher production disk filemode
2. SQL/DS user ID with DBA authority
3. SQL/DS connect password for the above user ID
4. Password for the DataRefresher user ID created by DVREXRPR
5. SQL/DS version and release number
6. SQL/DS database name

A message will be issued by DVREXBSSR when it is necessary to run DVREXRPR. Before running DVREXRPR, the following environment must be established:

- DataRefresher production disk accessed
- SQL/DS code disk accessed
- SQL/DS database started
- User ID that is running the DVREXRPR EXEC must have done an SQLINIT to the SQL/DS database

**DVREXRS1** Loads the DataRefresher sample data from the DataRefresher production disk to the SQL/DS database. To run this EXEC, four input parameters are required. Use the worksheet in Appendix B, “VM installation worksheet” on page 183 to help you gather these parameters. The parameters are positional and are as follows:

1. DataRefresher production disk filemode
2. SQL/DS user ID with DBA authority
3. SQL/DS connect password for the above user ID
4. Password for the DataRefresher user ID created by DVREXRPR

A message will be issued by DVREXBSSR when it is necessary to run DVREXRS1. Before running DVREXRS1, the following environment must be established:

- DataRefresher production disk accessed
- SQL/DS code disk accessed
- SQL/DS database started
- User ID that is running the DVREXRS1 EXEC must have done an SQLINIT to the SQL/DS database

**DVREXBMV** Creates the sample data MACLIB, DVRDATAE, for use with the DataRefresher dialogs. To run this EXEC, five input parameters are required. The worksheet in Appendix B, “VM installation worksheet” on page 183 will help you gather these parameters. The parameters are positional and are as follows:

1. The characters YES which indicate the Relational Data Extract feature is installed
2. DataRefresher production disk filemode
3. ISPF disk filemode
4. SQL/DS user ID with DBA authority
5. SQL/DS connect password for the above user ID
A message will be issued by DVREXBMR when it is necessary to run DVREXBMV. Before running DVREXBMV, the following environment must be established:

- DataRefresher production disk accessed
- ISPF code disk accessed
- SQL/DS code disk accessed
- SQL/DS database started
- User ID that is running the DVREXBMV EXEC must have done an SQLINIT to the SQL/DS database

---

**Installing the DataRefresher VM base product and the Relational Data Extract feature**

The following overviews summarize the tasks for installing the DataRefresher base product and the Relational Data Extract feature on a VM operating system.

**DataRefresher base product task overview**

The following is a task overview of the installation process for the DataRefresher base product. You can use the space at the left of each step to check it off when the step is completed.

- 1. Read the DataRefresher program directory
- 2. Complete the installation worksheet in Appendix B
- 3. Allocate disk space
- 4. Load the distribution and production minidisks from the product tape
- 5. Run the installation control EXEC
- 6. Run the installation verification program
- 7. If migrating from DXT, run the Master Index Table Migration Utility
- 8. Make DataRefresher dialogs an ISPF/PDF application

**Relational Data Extract feature task overview**

The following is a task overview of the installation process for the Relational Data Extract feature. You can use the space at the left of each step to check it off when the step is completed.

- 1. Read the DataRefresher program directory
- 2. Complete the installation worksheet in Appendix B
- 3. Control access to SQL/DS database machine
- 4. Load the distribution and production minidisks from the product tape
- 5. Run the installation verification program

**Note:** For full installation procedures, see the DataRefresher program directory.
After you install DataRefresher

The tasks you must perform to administer DataRefresher at your site will vary according to the DataRefresher configuration you have chosen. Besides what has been covered in this chapter, you might also need to administer the Relational Data Extract Feature and start the DataRefresher dialogs.

For more information, see the following:

- Chapter 14, "Relational Data Extract feature" on page 145
- Chapter 17, "Setting up VM dialogs" on page 165
Chapter 6. Installing the Uppercase and Kanji Features

The Uppercase Feature (UCF), when supported by a Katakana terminal, enables you to access to both Katakana and Latin characters. The Kanji Feature enables you to operate the DataRefresher dialogs in a Kanji language environment.

The UCF and the Kanji Feature can be installed on an MVS host system with the DataRefresher Installation Feature, either during or after DataRefresher base product installation. The UCF and the Kanji Feature can only be installed on a VM host system after the DataRefresher base product has been installed.

This chapter provides information you should know before you install and run the Uppercase and Kanji Features.

- "National Language Features and the DataRefresher dialogs environment" describes how to use an NLF in a DataRefresher dialogs environment.
- "NLF components" on page 59 describes what components are added to DataRefresher components when you install an NLF.
- "Using double-byte character sets (DBCS) data" on page 60 describes how to work with double byte character set (DBCS) data.
- "Preparing to install the Uppercase Feature on an MVS system" on page 60 describes requirements and provides task overviews for installing the Uppercase Feature on MVS.
- "Preparing to install the Kanji Feature on an MVS system" on page 62 describes requirements and provides task overviews for installing the Kanji Feature on MVS.
- "Preparing to install the Uppercase Feature on a VM system" on page 63 describes requirements and provides a task overview for installing the Uppercase Feature on VM.
- "Preparing to install the Kanji Feature on a VM system" on page 64 describes requirements and provides a task overview for installing the Kanji Feature on VM.

The DataRefresher program directory provides information on DataRefresher manuals currently available in translation.

National Language Features and the DataRefresher dialogs environment

A National Language Feature (NLF) provides DataRefresher dialog users with a session environment that is tailored to a specific language. The UCF and the Kanji Feature are both DataRefresher NLFs.

When you install one or more NLFS, your DataRefresher environment becomes a multilingual environment. In such an environment, users with the proper authorization may choose a language for each of their DataRefresher dialogs sessions. A user who has installed the Kanji feature, for example, could choose English for one session and Kanji for another.
The difference between the English and NLF environments lies not in what you can do, but in the language you see on your terminal screen. You can perform the same DataRefresher operations in an NLF session that you can during an English-language DataRefresher dialogs session. The DataRefresher command language is the same for both English and NLF environments.

The DataRefresher administrator determines which language environments are available to a user. For example, one user could be confined to English-language sessions, and another to Kanji-language sessions. You cannot switch languages during a DataRefresher dialogs session. Instead, to operate DataRefresher in a different language environment, you must end the current DataRefresher dialogs session and begin another.

Information translated from English for an NLF

Like the English-language environment, each NLF environment has a complete set of dialog panels and messages. In most cases, the NLF panels and messages are translated into the appropriate language. Only the fixed portions of text can be translated; variable information (extract names, for example) is not translated.

User profile parameter values are established during the first enrollment process. For example, if you were first enrolled for the English language environment, your profile defaults would be for English models. If you were later to enroll yourself in the Kanji language environment, your profile would still default to the English model names - until you used the Administrative Dialogs profile option to change the defaults to Kanji model names.

Profiles with different model defaults contain the same parameters. The names of the parameters, however, may be translated. Additionally, the values of certain parameters may be translated.

Information not translated from English for an NLF

The following are not translated from English for an NLF:

- Inline comments in DataRefresher dialogs models for batch commands (such as CREATE DTXFILE and EXTRACT) are not translated, nor are the commands.
- Dialogs commands.
- Sample data files.
- Commands run by the batch components are not translated. The batch components include the UIM, DEM, REM, DRU, and DAP.
- Batch component messages.
- DataRefresher dialogs command models and JCL modules.

Information translated for the Uppercase and Kanji Features

All input entered on dialogs panels must be entered in uppercase English, for both the Uppercase and Kanji Features.

- For the UCF, all fixed text in the panels and messages are translated to uppercase English, for use with terminals equipped with Katakana support (555x, 556x). Variable information, like column and table names, may or may not appear in the panels and messages in uppercase. Although
DataRefresher's command language is exactly the same for English and the UCF, every command must be entered entirely in uppercase for the UCF.

- For the Kanji environment, the DataRefresher command language is the same as it is for the UCF. Panels and end-user messages are translated into Kanji characters.

**NLF components**

Installing an NLF does not replace or modify existing DataRefresher components. Instead, the NLF components are simply added to the existing software. NLF components include several program modules and a number of partitioned data sets or libraries (MACLIBs). For every NLF component, there is a corresponding English-language component installed as part of the DataRefresher base product. DataRefresher uses the NLF components, rather than their English-language counterparts, when it runs DataRefresher sessions in an NLF environment.

**How NLF components work**

The NLF program modules contain language-sensitive DataRefresher code. When a user begins a DataRefresher session using an NLF, one of these modules receives control and then runs the DataRefresher code to conduct the session dialog.

The NLF libraries include ISPF panel and message libraries. These panels and message libraries are translated versions of the corresponding English-language library. For example, the members of the Kanji panel library are English-language panel definitione, with the fixed text translated into Kanji characters.

**Naming conventions for NLF components**

Each NLF is assigned a one-letter language identifier, and NLF component names and those of their English-language counterparts differ only in this single letter. For the program modules and libraries, this language identifier is the last letter in the name. For the library members (panel definitions, message modules, and so forth), the identifier is the fourth character in the name. Table 5 gives current DataRefresher language identifiers.

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>English</td>
</tr>
<tr>
<td>U</td>
<td>Uppercase</td>
</tr>
<tr>
<td>K</td>
<td>Kanji</td>
</tr>
<tr>
<td>X</td>
<td>non-translatable</td>
</tr>
</tbody>
</table>

When you see a module, library, or job containing a # (number sign), replace the # with the appropriate letter for the national language you are installing. Here are some examples in which the distinguishing letter is represented by a #.

- DXT310.DVRMLIB# is the name of the ISPF message library in MVS.
- DVRMLIB# MACLIB is the name of the ISPF message MACLIB in VM.
• DVRD000# is the name of the program module that initializes DataRefresher for the corresponding language environment.

• DVR#Hnnn is the panel definition for a message help panel.

Thus, DVRKH100 is the Kanji translation of the English-language panel DVREH100. The naming convention ensures that all library members have unique names, regardless of the libraries in which they reside.

Using double-byte character sets (DBCS) data

To enable you to use the Uppercase feature and Kanji feature, DataRefresher lets you enter, process and display double-byte character set (DBCS) data. The data you enter on a DBCS terminal is stored using two bytes for each character. The underlying database management system, IMS, DB2, or SQL/DS (character or graphic data type) provides the actual storage of the two-byte data. Individual fields or columns may contain only extended binary-coded decimal interchange code (EBCDIC) data, only DBCS data, or a mixture of DBCS and EBCDIC data.

DataRefresher lets you specify double-byte characters in extract requests so that you can extract particular DBCS data from pertinent databases. Also, you can use DBCS to enter the names and descriptions of many DataRefresher objects, such as:

• The extract request identification
• DXTVIEW names that identify the DataRefresher view description
• Alias names that are given to like-named fields in a DXTVIEW description

For information on entering DBCS data, see the DataRefresher Command Reference.

Preparing to install the Uppercase Feature on an MVS system

The DataRefresher base product distribution and target libraries are also used for the UCF.

Before installing the UCF on MVS, you must ensure that additional DASD space is available for the distribution libraries and the target libraries. The DataRefresher program directory contains information on additional DASD space requirements for UCF distribution and target libraries.

Distribution and target libraries for the UCF

Table 6 and Table 7 on page 61 describe the distribution and target libraries for the UCF:

<table>
<thead>
<tr>
<th>Table 6. Distribution library for the Uppercase Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCF Distribution Library</td>
</tr>
<tr>
<td>DVR110:ADVRPMSU</td>
</tr>
</tbody>
</table>
Table 7. Target libraries for the Uppercase Feature

<table>
<thead>
<tr>
<th>UCF Target Libraries</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVR110.DVRSAMPU</td>
<td>IVP, sample requests, procedures</td>
</tr>
<tr>
<td>DVR110.DVRPLIBU</td>
<td>ISPF panels</td>
</tr>
<tr>
<td>DVR110.DVRJEDIU</td>
<td>Sample JCL</td>
</tr>
<tr>
<td>DVR110.DVRMLIBU</td>
<td>Sample ISPF messages</td>
</tr>
</tbody>
</table>

Task overview

You can install the UCF during DataRefresher MVS/ESA installation or any time after DataRefresher has been installed at your site. You install the UCF on MVS using the DataRefresher Installation Feature. The following list summarizes the additional steps you must take during DataRefresher installation to install the UCF. You can use the space at the left of each step to check it off after you have completed it.

1. Retrieve files from the UCF ISD Tape
2. Allocate space for target and distribution libraries
3. Run SMP RECEIVE and APPLY for the UCF
4. Make DataRefresher dialogs an ISPF/PDF Application
5. Make DataRefresher CLISTs available to TSO
6. Enroll installer to be able to run the IVP
7. Run job to initialize the DataRefresher dialogs environment
8. Run the IVP
9. Run job to cleanup DataRefresher dialogs environment
10. Run SMP ACCEPT for the UCF

If DataRefresher is already installed at your site and you want to install the UCF:

1. Read the DataRefresher program directory
2. Complete the worksheets in Appendix A, “MVS installation worksheets” on page 177
3. Run the DataRefresher Installation Feature
4. When prompted to install the UCF, enter Y
5. Follow the steps outlined in the task overview above
Preparing to install the Kanji Feature on an MVS system

The Kanji Feature has special hardware and SMP requirements which must be met before it can be installed on MVS.

Hardware and software product requirements

The Kanji Feature requires ISPF Version 2 Release 2, and a workstation equipped to handle DBCS data, for example the IBM Personal System/55.

SMP requirements

In addition to the DASD space required for the base product, more DASD space is required for the distribution and target libraries. The figures are shown in the DataRefresher program directory. Any SMP data description names (DD names) that are not listed require less than 5 tracks per DD name.

Distribution and target libraries for the Kanji Feature

Tables 8 and 9 describe the distribution and target libraries for the Kanji Feature.

<table>
<thead>
<tr>
<th>Kanji Distribution Library</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVR110.ADVRPMSK</td>
<td>ISPF panels, Kanji Feature installation procedures, IVP, and DataRefresher procedures</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Kanji Target Libraries</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVR110.DVRSAMPK</td>
<td>IVP and procedures</td>
</tr>
<tr>
<td>DVR110.DVRPLIBK</td>
<td>ISPF panels</td>
</tr>
<tr>
<td>DVR110.DVRMLIBK</td>
<td>Sample ISPF messages</td>
</tr>
<tr>
<td>DVR110.DVRJEDIK</td>
<td>JCL</td>
</tr>
</tbody>
</table>

Task overview

You can install the Kanji Feature during DataRefresher MVS/ESA installation or any time after DataRefresher has been installed at your site. You install the Kanji Feature on MVS using the DataRefresher Installation Feature. The following list summarizes the additional steps you must take during DataRefresher installation to install the Kanji Feature. You can use the space at the left of each step to check it off after you have completed it.

- 1. Retrieve files from the NLF ISD tape
- 2. Allocate space for target and distribution libraries
- 3. Run SMP RECEIVE and APPLY for the Kanji Feature
- 4. Make DataRefresher dialogs an ISPF/PDF application
- 5. Make DataRefresher CLISTs available to TSO
- 6. Enroll installer to be able to run the IVP
- 7. Run job to initialize the DataRefresher dialogs environment
- 8. Run the IVP
- 9. Run job to cleanup DataRefresher dialogs environment
- 10. Run SMP ACCEPT for the Kanji Feature
If DataRefresher is already installed at your site and you want to install the Kanji Feature:

1. Read the DataRefresher program directory
2. Complete the installation worksheet in Appendix A, "MVS installation worksheets" on page 177
3. Run the DataRefresher Installation Feature
4. Enter Y when prompted to install the Kanji Feature
5. Follow the steps outlined in the task overview above

Preparing to install the Uppercase Feature on a VM system

You must first install the DataRefresher VM/ESA base product before you can install the DataRefresher UCF. The UCF uses the existing DataRefresher distribution disk and requires a separate production disk. For allocation size, see the DataRefresher program directory.

Table 10 describes the production libraries and files for the UCF.

<table>
<thead>
<tr>
<th>Production Libraries</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVRUxxxx EXEC</td>
<td>Installation procedures, where xxxx represents the installation procedures</td>
</tr>
<tr>
<td>DVRIO000 LOADLIB</td>
<td>Code to handle DataRefresher I/O operations</td>
</tr>
<tr>
<td>DVRDATAU MACLIB</td>
<td>The DataRefresher sample tables</td>
</tr>
<tr>
<td>DVRJEDIU MACLIB</td>
<td>The model extract requests, data description requests, and execution JCL</td>
</tr>
<tr>
<td>DVRMLIBU MACLIB</td>
<td>ISPF messages for running DataRefresher dialogs</td>
</tr>
<tr>
<td>DVRPLIBU MACLIB</td>
<td>ISPF panels for running DataRefresher dialogs</td>
</tr>
<tr>
<td>DVRSLIBE MACLIB</td>
<td>ISPF skeletons for running DataRefresher dialogs</td>
</tr>
<tr>
<td>DVRRTADME MACLIB</td>
<td>The ISPF table for commands</td>
</tr>
<tr>
<td>DVRTLIBU MACLIB</td>
<td>The table library for shared objects library</td>
</tr>
<tr>
<td>DVRLOADU TXTLIB</td>
<td>The text modules necessary to run DataRefresher dialogs</td>
</tr>
<tr>
<td>DVRQ000U TXTLIB</td>
<td>The text modules necessary to update the Master Index Table</td>
</tr>
<tr>
<td>DVRRT0000 TEXTLIB</td>
<td>The text modules necessary to migrate the Master Index Table</td>
</tr>
</tbody>
</table>
Task overview

The following list summarizes the tasks for installing the UCF. You can use the space at the left of each step to check it off after you have completed it.

- 1. Read the DataRefresher program directory
- 2. Complete the installation worksheet in Appendix B, "VM installation worksheet" on page 183
- 3. Allocate disk space
- 4. Load the distribution and production minidisks from the ISD tape
- 5. Run the installation EXEC
- 6. Run the EXEC to initialize DataRefresher dialogs environment and enroll installer
- 7. Make DataRefresher dialogs an ISPF/PDF application
- 8. Run the UCF DataRefresher dialogs IVP

Preparing to install the Kanji Feature on a VM system

You must first install the DataRefresher VM/ESA base product before you can install the Kanji Feature. The Kanji Feature has special hardware program and disk storage requirements which must be met before it can be installed on VM.

Hardware and program product requirements

The Kanji Feature requires:

- a workstation equipped to handle DBCS characters, such as the IBM Personal System/55
- ISPF Version 2 Release 2

In addition, make sure that your ISPF environment is set up to display the Kanji characters.

Disk storage requirements for Kanji Feature

The Kanji Feature uses the DataRefresher base distribution and production minidisks. However, an additional Kanji production minidisk is required. For allocation size, see the DataRefresher program directory.
Production libraries for the Kanji Feature

Table 11 describes the production libraries and files for the Kanji Feature.

<table>
<thead>
<tr>
<th>Production Libraries</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVRKxxxx EXEC</td>
<td>Installation procedures, where xxxx represents the installation procedures</td>
</tr>
<tr>
<td>DVRIO000 LOADLIB</td>
<td>Code to handle DataRefresher I/O operations</td>
</tr>
<tr>
<td>DVRDATAK MACLIB</td>
<td>The DataRefresher sample tables descriptions</td>
</tr>
<tr>
<td>DVRJEDIK MACLIB</td>
<td>The model extract requests, data description requests, and execution JCL</td>
</tr>
<tr>
<td>DVRMLIBK MACLIB</td>
<td>ISPF messages for running DataRefresher dialogs</td>
</tr>
<tr>
<td>DVARPLIBK MACLIB</td>
<td>ISPF panels for running DataRefresher dialogs</td>
</tr>
<tr>
<td>DVRSLIBE MACLIB</td>
<td>ISPF skeletons for running DataRefresher dialogs</td>
</tr>
<tr>
<td>DVRADME MACLIB</td>
<td>The ISPF table for commands</td>
</tr>
<tr>
<td>DVRTLIBK MACLIB</td>
<td>The table library for shared objects library</td>
</tr>
<tr>
<td>DVRLOADK TXTLIB</td>
<td>The text modules necessary to run DataRefresher dialogs</td>
</tr>
<tr>
<td>DVRQ000K TXTLIB</td>
<td>The text modules necessary to run the Master Index Table Migration Utility</td>
</tr>
<tr>
<td>DVRRT0000 TXTLIB</td>
<td>The text modules necessary to use the Master Index Table Migration Utility</td>
</tr>
</tbody>
</table>

Task overview

The following list summarizes the tasks for installing the Kanji Feature. You can use the space at the left of each step to check it off after you have completed it.

1. Read the DataRefresher program directory
2. Complete the installation worksheet in Appendix B, "VM installation worksheet" on page 183
3. Allocate disk space
4. Load the distribution and production minidisks from the ISD tape
5. Run the installation EXEC
6. Run the EXEC to initialize DataRefresher dialogs environment and enroll installer
7. Make DataRefresher dialogs an ISPF/PDF application
8. Run the Kanji DataRefresher dialogs IVP
Part 3. Administering DataRefresher

This part provides information on administering DataRefresher to optimize DataRefresher performance.

Chapter 7. General Data Extract feature
Task overview

Chapter 8. Preparing to use the General Data Extract feature
Extracting data using the General Data Extract feature
What you need
Running the UIM and DEM
Step 1: Creating data descriptions
Step 2: Submitting data descriptions to the UIM
Step 3: Building extract requests
Step 4: Submitting extract requests to the UIM
Step 5: Running extract requests
Planning to use single or multiple FDTLIBs
Preparing for Data Extract Manager operation
Preparing to use exit routines

Chapter 9. Setting up FDTLIB and EXTLIB data sets
Administering the FDTLIB
Estimating FDTLIB storage
Allocating FDTLIB storage
Reallocating FDTLIB storage
Administering the EXTLIB
Estimating EXTLIB storage
Allocating EXTLIB storage
Reallocating EXTLIB storage

Chapter 10. Setting up the Data Extract Manager (DEM)
Deciding what type of DEM to run
Setting up the DEM environment
Setting the run mode and polling interval for a DEM
Setting up extract request priorities
Setting defaults for dynamic allocation
Setting up DEM output limits
Determining diagnostic output requirements
Determining which extract requests a DEM processes
Information necessary to write DEM JCL and initialization commands
Specifying source data sets for DEM extract requests
Non-IMS MVS batch job
IMS batch job
IMS BMP job
Specifying output data sets for DEM extract requests
Batching extract requests
Batching high-priority extract requests
Determining which extract requests to batch
Setting up a message table
Administering DEM
Communicating with a DEM
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Displaying DEM information</td>
<td>113</td>
</tr>
<tr>
<td>Stopping a DEM</td>
<td>115</td>
</tr>
<tr>
<td>Changing a DEM's operating characteristics</td>
<td>117</td>
</tr>
<tr>
<td>Recovering from a DEM error</td>
<td>120</td>
</tr>
<tr>
<td>Recovering from DXTOUT data set WRITE errors</td>
<td>122</td>
</tr>
<tr>
<td><strong>Chapter 11. Protecting your General Data Extract data</strong></td>
<td>123</td>
</tr>
<tr>
<td>Protecting individual data descriptions</td>
<td>123</td>
</tr>
<tr>
<td>Using DVRDEFN to protect data</td>
<td>125</td>
</tr>
<tr>
<td>Protecting data descriptions created by the DAP</td>
<td>128</td>
</tr>
<tr>
<td>Controlling access to the FDTLIB and EXTLIB data sets</td>
<td>129</td>
</tr>
<tr>
<td>Protecting FDTLIBs</td>
<td>129</td>
</tr>
<tr>
<td>Protecting EXTLIBs</td>
<td>129</td>
</tr>
<tr>
<td>Preventing unauthorized use of the UIM and DEM</td>
<td>130</td>
</tr>
<tr>
<td>Data-protection schemes for single or multiple FDTLIBs</td>
<td>131</td>
</tr>
<tr>
<td>Using a single FDTLIB</td>
<td>131</td>
</tr>
<tr>
<td>Using multiple FDTLIBs</td>
<td>132</td>
</tr>
<tr>
<td>Protecting source data</td>
<td>133</td>
</tr>
<tr>
<td>Protecting exit routines</td>
<td>133</td>
</tr>
<tr>
<td>Controlling DataRefresher OS/2 access to host resources</td>
<td>134</td>
</tr>
<tr>
<td>Host security</td>
<td>134</td>
</tr>
<tr>
<td>APPC security</td>
<td>134</td>
</tr>
<tr>
<td><strong>Chapter 12. Administering the Dictionary Access Program</strong></td>
<td>135</td>
</tr>
<tr>
<td>What the Dictionary Access Program does</td>
<td>135</td>
</tr>
<tr>
<td>Writing input for the DAP</td>
<td>136</td>
</tr>
<tr>
<td>Writing JCL to run the DAP</td>
<td>136</td>
</tr>
<tr>
<td>Reviewing the DAP output</td>
<td>136</td>
</tr>
<tr>
<td>What information the DAP needs from the data dictionary</td>
<td>137</td>
</tr>
<tr>
<td>How the DAP determines the values of NAME keywords</td>
<td>137</td>
</tr>
<tr>
<td>How the DAP determines the values of other keywords</td>
<td>138</td>
</tr>
<tr>
<td>What the DAP retrieves for the DESC keyword</td>
<td>139</td>
</tr>
<tr>
<td>What the DAP retrieves for the FREQ keyword</td>
<td>139</td>
</tr>
<tr>
<td>How the DAP interprets the GENFLD value</td>
<td>139</td>
</tr>
<tr>
<td>How the DAP interprets an alternate PCB</td>
<td>140</td>
</tr>
<tr>
<td>Possible conflicts in dictionary data usage</td>
<td>140</td>
</tr>
<tr>
<td><strong>Chapter 13. Managing extract requests</strong></td>
<td>141</td>
</tr>
<tr>
<td>Choosing a method for running an extract on a regular basis</td>
<td>141</td>
</tr>
<tr>
<td>Managing data sets containing JCL, JCS, and DataRefresher commands</td>
<td>142</td>
</tr>
<tr>
<td>Creating partitioned data sets</td>
<td>142</td>
</tr>
<tr>
<td>Grouping data sets according to frequency of use</td>
<td>143</td>
</tr>
<tr>
<td>Updating the EXTLIB with extract requests</td>
<td>143</td>
</tr>
<tr>
<td><strong>Chapter 14. Relational Data Extract feature</strong></td>
<td>145</td>
</tr>
<tr>
<td>Task overview</td>
<td>145</td>
</tr>
<tr>
<td>Implementing a data-protection scheme for the Relational Data Extract feature</td>
<td>146</td>
</tr>
<tr>
<td>Controlling access to data with relational database views</td>
<td>146</td>
</tr>
<tr>
<td>Granting and revoking data access with SQL authorization statements</td>
<td>146</td>
</tr>
<tr>
<td>Using CONNECT authority to control access to an SQL/DS database</td>
<td>147</td>
</tr>
<tr>
<td><strong>Chapter 15. Setting up online commands</strong></td>
<td>149</td>
</tr>
<tr>
<td>Task overview</td>
<td>149</td>
</tr>
<tr>
<td>Using DataRefresher online commands with DataRefresher features</td>
<td>149</td>
</tr>
</tbody>
</table>
Chapter 16. Setting up MVS dialogs

Task overview ........................................ 157
Using tables and data sets from DXT dialogs .......... 157
  DXT dialog components that are compatible with DataRefresher ... 157
  Renaming DXT dialog user data sets ............... 158
  Using an ISPF profile data set (ISPPROF) ........... 158
Editing CLISTs for your environment ..................... 158
  Editing DVRELNRCLIST ................................ 159
  Editing DVRELLGN ..................................... 160
  Setting up the end user environment ............... 161
Enrolling the administrator ................................ 161
  Calling the DVRELNRCLIST CLIST ................. 161
  How the CLIST runs .................................. 162
DataRefresher dialogs models ............................ 162
  Data description models .............................. 162
  Extract request models ................................ 163
  JCL and JCS models .................................. 163
  SAP skeletons ....................................... 163
Maintaining library space for your users ............... 164
Administering the End User Dialogs ..................... 164
  Building the nickname table ....................... 164
  Updating the end user table ....................... 164
  Requesting source descriptions ................. 164

Chapter 17. Setting up VM dialogs

Task overview ........................................ 165
Using tables and MACLIBs from DXT dialogs ............ 165
  DXT dialog components that are compatible with DataRefresher ... 165
  Using DXT dialog user MACLIBs .................... 166
  Using an ISPF profile MACLIB (ISPPROF) ........... 166
Editing VM EXECs for your environment ................. 166
  Editing DVREXLIB .................................... 167
  Editing DVREXUSR ................................... 167
  Setting up the end user environment ............... 168
  Editing DVREXSND .................................... 169
Enrolling the administrator ............................. 169
  Calling the DVREXLIB EXEC ....................... 170
  How the EXEC runs .................................. 170
DataRefresher dialogs models ............................ 170
  Data description models .............................. 171
  Extract request models ................................ 171
  JCL/JCS models ..................................... 171
  SAP Skeletons ....................................... 172
Maintaining library space for your users ............... 172
Administering the End User Dialogs ..................... 172
  Building the nickname table ....................... 172
  Updating the end user table ....................... 172
  Requesting source descriptions ................. 172
Chapter 7. General Data Extract feature

This chapter provides you with an outline of how to administer the General Data Extract feature.

Task overview

Following is a list of the administrative tasks you need to complete for the General Data Extract feature. Each task is briefly described, and references are provided to chapters in this book or to other books in the DataRefresher library where you can look for more information.

1. Prepare to use the General Data Extract feature.

   Before using the General Data Extract feature, you should have a basic understanding of the following:
   
   - Creating data descriptions
   - Building and submitting extract requests
   - Planning for single or multiple FDTLIBs
   - Making exit routines available

   For an introduction to General Data Extract feature operations, see Chapter 8, “Preparing to use the General Data Extract feature” on page 75.

2. Create and maintain FDTLIB and EXTLIB data sets.

   After DataRefresher is installed, you must allocate the storage areas for the data description library (FDTLIB) and the extract request library (EXTLIB).
   Also, you may have to enlarge or reduce these data sets to accommodate the needs of your site. To create and maintain FDTLIB and EXTLIB space, you must estimate the amount of storage area you need, and then run an OS/VS Access Method Services job to define or redefine the data sets.

   For information on working with FDTLIB and EXTLIB data sets, see Chapter 9, “Setting up FDTLIB and EXTLIB data sets” on page 87.

3. Set up Data Extract Manager operation.

   After the UIM saves the data descriptions in the FDTLIB and places a user's extract request in the EXTLIB, the DEM must be called to process the extract request. As DataRefresher administrator, you can assist in creating a suitable DEM operating environment by providing operator education, formulating scheduling rules, and setting up the JCL for individual runs.

   For information on administering the DEM, see Chapter 10, “Setting up the Data Extract Manager (DEM)” on page 97.

4. Install and customize DataRefresher OS/2 (optional).

   DataRefresher OS/2 enables your users to create data descriptions, views, and extract requests in a workstation environment. Extract requests created with DataRefresher OS/2 can be sent to the host MVS system for processing. For information on installing and customizing DataRefresher OS/2 for use at your site, see the DataRefresher OS/2 User's Guide.
5. Implement a data-protection scheme.

You can use the Resource Access Control Facility (RACF) to protect your data descriptions, data sets and DataRefresher resources from unauthorized use. Additionally, you can decide to use either a single FDTLIB or multiple FDTLIBs to control access to the data descriptions you have in storage. If you are operating DataRefresher OS/2 at your site, workstation access to data and resources on the host system can be provided through Advanced Program-to-Program Communications (APPC) security.

Administrative schemes for controlling access to your data and DataRefresher resources are discussed in Chapter 11, “Protecting your General Data Extract data” on page 123.

6. Create and maintain data descriptions.

As DataRefresher administrator, you must create and maintain data descriptions to enable users to access non-relational data structures, including partial paths, repeating groups, and variable-length data. This task includes adding, deleting, and replacing data descriptions using the UIM. For information on creating and maintaining data descriptions and submitting them to the UIM, see the DataRefresher MVS and VM User's Guide.


The DAP works with the IBM OS/VS DB/DC Data Dictionary to create data descriptions to submit to the UIM. You can use the DAP to simplify the creation and maintenance of data descriptions. For information on how to administer the DAP, see Chapter 12, “Administering the Dictionary Access Program” on page 135.

8. Administer the Structures Access Program (SAP).

The SAP works with IMS structures, physical sequential files, and high-level language libraries to create data descriptions to submit to the UIM. You can use the SAP to simplify the creation and maintenance of data descriptions. For information on how to administer the SAP, see the DataRefresher MVS and VM User's Guide.

9. Edit DataRefresher models to suit your site's requirements.

DataRefresher provides models of the following for your use:

- Data descriptions
- Extract requests
- JCL
- JCS

You can edit these models and make them available to your users to save time and reduce the chance for errors. For information and examples of how to edit these models, see the DataRefresher MVS and VM User's Guide.

10. Prepare to run repeated extract requests.

You may need to run the same extract requests on a regular basis to satisfy your data processing requirements. For information on setting up your environment to run repeated extracts, see Chapter 13, “Managing extract requests” on page 141.
11. Write DataRefresher exit routines.

You can write the following DataRefresher exit routines to extend the range of DataRefresher performance at your site:

- Accounting exit routines
- Data exit routines
- Date/Time Conversion exit routines
- User Data Type exit routines
- Generic Data Interface (GDI) exit routines
- Generic Output Interface (GOI) exit routines
- Map Capture exit routines

For a summary of the functions exit routines can perform, see “Preparing to use exit routines” on page 84.

For information on writing and using exit routines, see the DataRefresher Exit Routines.

12. Administer the DataRefresher online commands.

If you use DataRefresher in the online environment, you can tailor that environment by changing the list of DataRefresher online commands keyword defaults, thus simplifying and shortening keyword input.

For information on administering DataRefresher online commands, see Chapter 15, “Setting up online commands” on page 149.

13. Administer the DataRefresher dialogs.

If you use DataRefresher dialogs at your site, you can tailor each user’s dialogs environment. By tailoring DataRefresher dialogs, you reduce the number of decisions that users have to make, and so simplify DataRefresher use.

For information on starting the DataRefresher dialogs, see Chapter 16, “Setting up MVS dialogs” on page 157. For information on using the DataRefresher dialogs, see the DataRefresher MVS and VM User’s Guide.
Chapter 8. Preparing to use the General Data Extract feature

This chapter provides you with steps for using and administering the General Data Extract feature.

- "Extracting data using the General Data Extract feature" describes general steps for using the General Data Extract feature.
- "Planning to use single or multiple FDTLIBs" on page 80 describes administrative schemes for the FDTLIBs at your site.
- "Preparing for Data Extract Manager operation" on page 82 describes what you need to consider before setting up a DEM at your site.
- "Preparing to use exit routines" on page 84 describes the exit routines supported by DataRefresher, and suggests how they can be used to extend DataRefresher operation at your site.

Extracting data using the General Data Extract feature

This section describes the steps for using the General Data Extract feature to extract data on a host system. For information on extracting data using a workstation, see the DataRefresher OS/2 User's Guide.

What you need

To extract data using the General Data Extract feature, you need an installed User Input Manager (UIM) and Data Extract Manager (DEM). You also need JCL to route the extract request to the appropriate UIM or DEM, and JCS to route the output to the required destination.

Also, if your source data is described using the IBM OS/VS DB/DC Data Dictionary, you must have access to the Dictionary Access Program (DAP) while your data descriptions are being extracted from the data dictionary. If your source data is described using IMS database definitions, or COBOL or PL/I structures, you can use the Structures Access Program (SAP) to extract your data descriptions.

For information on writing JCL and JCS, and using the DAP and SAP, see the DataRefresher MVS and VM User's Guide.

Running the UIM and DEM

Extracting data using the UIM and DEM is a two-stage process:

1. You call the UIM to store your data descriptions and extract request
2. You call the DEM to run your extract request

You need to run the UIM to:

- Submit data descriptions. The UIM places descriptions of your source data into the FDTLIB. The DEM uses your data descriptions in the FDTLIB to process your extract request.
- Maintain data descriptions. You can print, punch, or delete existing data descriptions.
• Submit extract requests. The UIM places your extract requests in the EXTLIB.
• Maintain extract requests. You can list, check the status of, or cancel your extract requests.

You need to run the DEM to extract your data. The DEM gets your data descriptions from the FDTLIB and your extract request from the EXTLIB, and it uses this information to extract the data and move it to the specified target.

Extracting data using both the UIM and DEM involves the following steps:

1. Creating data descriptions
2. Submitting data descriptions to the UIM
3. Building extract requests
4. Submitting extract requests to the UIM
5. Running extract requests

Step 1: Creating data descriptions

A data description describes to DataRefresher the structure of a data set or IMS database from which you want to extract either a subset of data or all of the data. DataRefresher lets you extract from a variety of non-relational data structures, including:

• Partial paths
• Repeating groups
• Variable-length data

The types of DataRefresher data descriptions include:

**DXTFILE description**

A DXTFILE description describes sources from which the DEM can extract data, including:

• VSAM data sets
• Physical sequential data sets
• Generic Data Interface (GDI) sources (databases or files)

DXTFILE descriptions tell DataRefresher the structure of the files; these files can be simple files (one record type) or structured files (multiple record types or internal segments). A DXTFILE description also includes information about the organization of the file (for example, physical sequential), and describes the fields within the file’s logical records or segments.

**DXTPSB description**

The DXTPSB description describes an IMS program specification block (PSB) through which the DEM can extract data from one or more IMS databases.

This DXTPSB contains DXTPCB descriptions, which map to the IMS program communication blocks (PCBs). The IMS PCBs contain information about which database a program is to use, the segment types within the database which that program uses, and which processing options are required by the program.

The DXTPCBs describe the segments and fields within the IMS PCBs. A segment description includes such items as its length, format (variable or fixed-length), and the name of the parent segment. If an IMS
database segment contains internal segments (repeating groups), these are defined to DataRefresher as segments. A field description includes such items as its length, location in the segment, and data characteristics.

You should set up any DXTPCBs with the assistance of the IMS database administrator at your site.

**DXTVIEW description**

The DXTVIEW description defines a view of a DXTFILE or DXTPSB. The data descriptions of a DXTFILE or DXTPSB provide the basis for defining the DXTVIEW.

A DXTVIEW describes only those segments and fields in a single path of the hierarchy that a user can retrieve.

As a DataRefresher administrator, you create a DXTVIEW, and select those fields defined in the DXTFILE (for non-IMS files) or DXTPSB (for IMS databases) that a user may access. You can create multiple DXTVIEWs for a DXTFILE or a DXTPSB.

DataRefresher automatically generates dummy views (no fields are defined) for GDI Select files.

**DataRefresher user data type description**

A DataRefresher user data type description defines a user data type. A user data type can be any data format not supported by DataRefresher. A user data type can only be processed with a User Data Type exit routine. You can reference these DataRefresher user data type descriptions in either a DXTFILE or DXTPSB description that you create.

For information on writing and using User Data Type exit routines, see the *DataRefresher Exit Routines*.

You need to submit your DataRefresher data descriptions to the UIM only when you first create them, and whenever the descriptions change.

**Creating data descriptions using DataRefresher CREATE commands**

You can create data descriptions using the following DataRefresher CREATE commands:

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

For information on using DataRefresher CREATE commands, see the *DataRefresher MVS and VM User’s Guide*.

**Creating data descriptions using the Dictionary Access Program**

If your site has the IBM OS/VS DB/DC Data Dictionary, you can simplify the job of creating DXTFILE and DXTPSB descriptions by using the data dictionary with the DataRefresher Dictionary Access Program (DAP).

After you identify files and PSBs to the DAP, the DAP, working with the data dictionary, creates data descriptions for the designated files and PSBs. The DAP
gives you an alternative to creating data descriptions and helps reduce the chance for error in coding a data description.

You can provide the input—the command to invoke the DAP and the JCL to run the DAP—as part of your own batch job or through the Administrative Dialogs.

For more information on using and administering the DAP, see the DataRefresher MVS and VM User's Guide.

**Step 2: Submitting data descriptions to the UIM**

After you create the necessary data descriptions, you must submit them to the UIM so they can be validated and later used by the DEM to extract your data. The UIM saves these data descriptions in the FDTLIB, as shown in Figure 17.

![Diagram](image)

*Figure 17. The FDTLIB stores your data descriptions*

You can submit these data descriptions using your own batch job, or the Administrative Dialogs, or you can process them via the DataRefresher online commands.

For a DXTVIEW description to be validated successfully, the DXTFILE or DXTPSB named in the DXTVIEW must be in the FDTLIB. If the FDTLIB does not contain the data description of a given database or file, you cannot extract data from that database or file.

Your data descriptions will remain in the FDTLIB and can be used repeatedly. If your source data changes, you will need to update your data descriptions.

**Step 3: Building extract requests**

You must submit an extract request to the UIM for each extract you want to do. You can build and submit your extract request using one of the following methods:

- The DataRefresher dialogs
- The DataRefresher online commands
- Your own batch job

DataRefresher lets you use a subset of the SQL SELECT statement as the extract language for DEM extract requests. Your DEM extract request has two parts:

1. The first part, the SUBMIT command, specifies information about the extract request and some of its output characteristics to the DEM. This command:
   - Names the extract request
   - Specifies the extracted output data target
   - Sends messages to the appropriate place
   - Specifies the format of the extracted data
   - Specifies DEM accounting information
   - Specifies exit routine information
2. The second part, the EXTRACT statement, tells the DEM what data you want to extract. This statement:
   - Specifies what data you want to extract
   - Provides information (such as an extract request's priority) that partially controls the DEM's processing of your extract request
   - Specifies the target relational database table, if appropriate

Step 4: Submitting extract requests to the UIM
After you build an extract request and submit it to the UIM, the UIM validates it and sends it to the EXTLIB, as shown in Figure 18.

![Diagram](image)

*Figure 18. The EXTLIB stores your extract requests*

The extract request is queued in the EXTLIB and remains there until the user cancels it or the DEM removes it. An extract request that has been processed or canceled is removed from the EXTLIB by default after seven days have passed, unless it has been defined as persistent. For information on running persistent extracts, see “Choosing a method for running an extract on a regular basis” on page 141.

Successful processing of the extract request depends on the data description already in place.

Step 5: Running extract requests
After the UIM saves the data descriptions in the FDTLIB and places the extract request in the EXTLIB, a user who has authority to run the DEM at your site must run the DEM to process the extract request, using either a batch job or the DataRefresher online commands. The DEM gets your data descriptions from the FDTLIB and your extract request from the EXTLIB, and uses this information to extract your data and move it to the specified target, as shown in Figure 19 on page 80.
Figure 19. The DEM processes your extract request

For more information about creating data descriptions and building, submitting and running extract requests on the host system, see the DataRefresher MVS and VM User's Guide.

For information on using DataRefresher OS/2 to create and submit views and extracts, see the DataRefresher OS/2 User's Guide.

Planning to use single or multiple FDTLIBs

To control access to your data descriptions, you can choose to use either a single FDTLIB or multiple FDTLIBs.

Using a single FDTLIB

If you choose to implement an administration scheme in which your site uses a single FDTLIB, a single DataRefresher administrator is responsible for the following:

- Protecting source data
- Creating and maintaining data descriptions, JCL files, and JCS files
- Building and maintaining extract requests (optional)

This type of administration scheme can provide for maximum control over your DataRefresher resources.

You can use this type of administration scheme if your site has any of the following characteristics:

- Users operating DataRefresher via End User Dialogs
- Users that regularly run the same extract request
- Central database tables that are updated regularly

Using multiple FDTLIBs

If you choose to implement an administration scheme in which your site uses multiple FDTLIBs, multiple DataRefresher administrators can perform part of the overall administration task and maintain personal DataRefresher dialogs libraries if the DataRefresher dialogs are used. This administration scheme allows minimum control and distributes the data protection and other administration tasks among several user groups. You can use this type of administration scheme if your site has the following characteristics:
• Experienced data-processing users
• Users who create their own data descriptions, JCL files, and JCS files
• Users who update existing database tables or have authority to create new ones

**Example using multiple FDTLIBs:** The figures for this example show one of many possible variations of using multiple FDTLIBs at your site.

For purposes of this example, assume that:

• User 1 and User 2 are in different departments.
• To protect data, each department has its own FDTLIB to which only department members have access.
• All departments at this site place their extract requests in the same EXTLIB.
• User 1 submits a high-priority extract request with small output; User 2 submits a low-priority extract request; both requests are submitted during peak hours.
• There are two DEMs, DEM1 and DEM2, set up for this site.
  – DEM1 is set up to run as quickly as possible all high-priority extract requests with small output submitted during peak hours.
  – DEM2 is set up to run all remaining extract requests during off-peak hours.

Thus, as Figure 20 shows, User 1 and User 2 can call the UIM at different times and direct their data descriptions to their respective FDTLIBs, but their extract requests will be stored in a single EXTLIB.

![Diagram showing storing data descriptions in multiple FDTLIBs](image)

*Figure 20. Storing data descriptions in multiple FDTLIBs*

After the UIM saves the data descriptions in the FDTLIBs and places the extract requests in the EXTLIB, the DEM must be called to run extract requests for User 1 and User 2.
As shown in Figure 21, DEM1 and DEM2 will be called at different times to run the extract requests for User 1 and User 2.

![Diagram showing DEM1 and DEM2 running extract requests](image)

**Figure 21. Running extracts with multiple FDTLIBs and multiple DEMs**

As Figure 21 shows, User 1’s high-priority extract request (JOB1) is run by DEM1, and accesses data set dsn1. User 2’s extract request (JOB2) is run by DEM2, and accesses data set dsn2.

---

**Preparing for Data Extract Manager operation**

At most sites, a DataRefresher user who submits an extract request does not run the DEM to process the extract request. However, the task of running the DEM can be performed by any DataRefresher user who can control the DEM. As a DataRefresher administrator, you can help set up the DEM operating environment by providing operator education, formulating scheduling rules, and setting up the necessary JCL for individual runs.

DEM operation consists of either of the following:

- Running a single DEM
- Running multiple DEMs at the same time

**Running a single DEM**

Before you can run a DEM, you must decide:

- Whether the DEM will run as:
  - An online DEM in the TSO foreground
  - A non-IMS MVS batch job
  - An IMS batch job
  - An IMS batch message processing (BMP) job
  - A batch job under CICS
- Which extract requests the DEM should process
- How much diagnostic information the DEM should print
- How you should schedule the DEM
- Whether the DEM can process extract requests in batches (if so, how many requests per batch)
Using DataRefresher commands, you can determine which extract requests are currently in the EXTLIB and the characteristics of those extract requests, what priorities the extract requests have, and what files or PSBs the extract requests use.

DataRefresher gives you the operational flexibility to run DEM extract requests as soon as you submit them, or later during off-peak hours. That is, you can set up DEM operation so that the DEM is run each time an extract request is submitted to the UIM, or so that the DEM is run later, during off-peak hours, after most users have submitted their extract requests to the UIM.

After the DEM is called, an MVS operator can control DEM processing, and has the option of intervening during DEM processing.

**Running multiple DEMs at the same time**

As Figure 22 indicates, you can have multiple DEMs running at the same time under one MVS operating system.

![Diagram of DEM operation](image)

*Figure 22. Running multiple DEMs under one MVS operating system*

Your DEMs can be set up to process only those extract requests with certain specifications. You must write and submit unique JCL for each DEM. However, one MVS operator can control more than one DEM.

For more information about administering the DEM, see Chapter 10, "Setting up the Data Extract Manager (DEM)" on page 97.
Preparing to use exit routines

You can further expand the power and flexibility of DataRefresher by using exit routines. Exit routines are user-written programs that enable you to accomplish a range of tasks that DataRefresher does not ordinarily perform. If you specify an exit routine in your data description or extract request, DataRefresher passes control to that exit routine at one or more points in the extract process. Once the exit routine completes its tasks, it returns control to DataRefresher, which continues processing your extract request.

The following types of exit routines can be used with the General Data Extract feature:

Accounting exit routines
Accounting exit routines enable you to monitor DEM usage at your site. For example, you can use an Accounting exit routine to identify authorized DEM users, or to keep track of resource utilization by particular users or accounts. Accounting exit routines can also be written to change the priority or the output limits of an extract request.

Data exit routines
Data exit routines enable you to change, manipulate, decode, or summarize source data that has been read but not yet examined by DataRefresher, to enable DataRefresher to process it. For example, your data may be coded or structured in unusual ways that do not conform to the data description in the FDTLIB. You can use a Data exit routine to interpret the data and transform it into a format (described in the data description) that DataRefresher can process.

Date/Time Conversion exit routines
Date/Time Conversion exit routines enable you to reformat date and time data from a user-defined format into the format set by the International Standards Organization (ISO), which is supported by DataRefresher. ISO date and time formats are shown in Table 12.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>yyyy-mm-dd</td>
<td>Always a 10-byte character string</td>
</tr>
<tr>
<td>TIME</td>
<td>hh:mm:ss</td>
<td>Always an 8-byte character string</td>
</tr>
</tbody>
</table>

Table 12. ISO date/time formats

where:

- yyyy = year
- mm = month
- dd = day
- hh = hours
- mm = minutes
- ss = seconds
DataRefresher supplies its own Date/Time Conversion exit routine to convert certain types of date and time formats to ISO. DataRefresher converts date and time data in the following formats automatically:

**Date formats:**
- mm/dd/yyyy (USA format)
- dd.mm.yyyy (EUR format)
- mm/dd/yy (local format)
- mm-dd-yy (local format)

**Time formats:**
- hh:mm xM where x=A or P (USA format)
- hh:mm:ss (JIS format)

Time data in EUR format (hh.mm.ss.) is the same as ISO and does not need to be converted.

If your source data contains date or time data that is neither in ISO format, nor in a format that DataRefresher converts automatically, you must write your own Date/Time Conversion exit routine before DataRefresher can process that data.

Date/Time Conversion exit routines do not support the conversion of timestamp data. If necessary, timestamp data can be converted with a User Data Type exit routine.

**User Data Type exit routines**
User Data Type exit routines enable you to transform specific fields in your source data from user-defined format into a format supported by DataRefresher. For example, User Data Type exit routines can be used to convert:

- Bit data
- Unsigned packed numeric data
- Three-byte binary data
- Encoded data (such as a 2-byte state ID that you want expanded)
- Numeric data stored as character data

User Data Type exit routines should be thought of as field level exit routines, in contrast to Data exit routines, which should be thought of as segment or record level exit routines.

**Generic Data Interface (GDI) exit routines**
GDI exit routines enable you to:

- Access otherwise unsupported IBM and non-IBM data sources
- Extract data from self-defining files, such as IXF files
- Join relational data sources with non-relational data sources
- Conserve system resources by doing two-stage extracts from DB2 databases

GDI exit routines are different from the other exit routines because it is the GDI exit routine, and not DataRefresher, that obtains the source data. DataRefresher calls the exit routine, the exit routine reads the data, and passes the data to DataRefresher.

There are two types of GDI exit routines:

- **GDI Select exit routines** are given the SELECT statement from your extract request, and return only the columns and rows of data
that satisfy the SELECT statement. With GDI Select exit routines, you have full SQL SELECT statement capability.

- **GDI Record exit routines** act as a user-written access method. They return records from any user-specified location to DataRefresher. DataRefresher applies the WHERE clause criteria you specified in your extract request and selects the appropriate fields. DataRefresher can join these records with data from other sources such as DL/I or VSAM data.

**Generic Output Interface (GOI) exit routines**

GOI exit routines enable you to manipulate data that has been extracted by DataRefresher and convert it into a user-defined format. For example, GOI exit routines enable you to:

- Format data for other databases
- Generate totals for extracted data
- Prepare data for downloading to a workstation database management system (DBMS)
- Summarize extracted data

After converting the data, GOI exit routines can do whatever you require with the data, such as write it to a file or immediately load it into a database.

**Map Capture exit routines**

Map Capture exit routines provide you with DataRefresher definition and extract request information so that you can save this information, or use it to perform additional checking. For example, your Map Capture exit routine could verify that the DB2 table you are extracting to has the correct table name, column names, and data types. Map Capture exit routines can also be used to capture information required for further processing by other programs, such as IBM's DataPropagator NonRelational.

You can code your exit routines in Assembler, COBOL, or PL/I. When writing JCL to run a DEM and UIIM, you need to make available libraries which contain language load modules and your exit routines. Included with DataRefresher are sample exit routines and interface control blocks for each type of exit routine.

For information on writing and using exit routines, see the *DataRefresher Exit Routines*.

For information on including your exit routines in extract requests submitted via DataRefresher OS/2, see the *DataRefresher OS/2 User's Guide.*
Chapter 9. Setting up FDTLIB and EXTLIB data sets

This chapter provides information you need before you can create and maintain your FDTLIB and EXTLIB data sets.

- “Administering the FDTLIB” provides formulae you can use to estimate storage for your FDTLIB, and example jobs you can use for allocating or reallocating FDTLIB storage.

- “Administering the EXTLIB” on page 92 provides a formula you can use to estimate storage for your EXTLIB, and example jobs you can use for allocating or reallocating EXTLIB storage.

Administering the FDTLIB

When DataRefresher is installed at your site, your system programmer runs procedures to test the installation. These procedures allocate storage areas for your data description library (FDTLIB) and your extract request library (EXTLIB). When the DataRefresher installation is complete, these library storage areas are automatically deleted.

Therefore, following installation, you must:

- Determine the approximate size of the FDTLIB and EXTLIB that your site needs to store the data descriptions and extract requests that need to be processed.

- Estimate the storage that you need for each library, and write an Access Method Services job to establish or change the size of the FDTLIB and the EXTLIB.

As your site's use of DataRefresher increases, you might decide to reallocate your FDTLIB to meet increased demand:

- As the users at your site access more databases, more data descriptions are added to your FDTLIB. Consequently, you need more space in your FDTLIB to store them.

- The contents of your FDTLIB can increase even if you do not access more databases, because you can create more than one DXTVIEW description for a single DataRefresher data description.

Note: In DataRefresher in an OS/2 environment, each FDTLIB and EXTLIB is represented by a DataRefresher World. You need to create a FDTLIB and EXTLIB, for each DataRefresher World using in DataRefresher under OS/2. This chapter only describes how to set these libraries up on the host system. For information about the DataRefresher World see the DataRefresher OS/2 User's Guide.
Estimating FDTLIB storage

To create or reallocate an FDTLIB, you must estimate the size of the descriptions that reside in the library. This section gives you formulae to help you determine the amount of space you need to store your data descriptions.

**Note:** For all formulae in this section,
- the '*' symbol is the multiply function
- the '/' symbol is the divide function

When dividing, use only the integral part of the quotient, and discard any remainder.

An FDTLIB has the following types of records for which you need to allocate space:

- DataRefresher user data type description records
- DXTFILE description records
- DXTPSB description records
- DXTVIEW description records
  - Internal DXTVIEW description records
  - Source DXTVIEW description records

A single FDTLIB record is of variable length and you can estimate the range from a minimum of 128 bytes to a maximum of 2000 bytes. However, a data description (which can contain up to 255 physical records) may span multiple FDTLIB records if the FDTLIB record is not large enough to contain the data description.

**Estimating storage for DataRefresher user data type descriptions**

Use the following formula to estimate the storage requirements in bytes for all your DataRefresher user data type descriptions:

\[ \text{number of bytes} = D \times 128 \]

where \( D \) is the number of DataRefresher user data type descriptions.

**Estimating storage for DXTFILE descriptions**

Use the following formula to estimate the storage requirements in bytes for one DXTFILE description:

\[ \text{number of bytes} = ((S+F) \times 266) + (((S+F)+5)/6) \times 272 \]

where \( S \) is the number of segments in a DXTFILE description, and \( F \) is the number of fields in a DXTFILE description.

**Estimating storage for DXTPSB descriptions**

Use the following formula to estimate the storage requirements in bytes for one DXTTop within a DXTPSB description. (Each PCB in a PSB starts a new record.)

\[ \text{number of bytes} = ((S+F) \times 266) + (((S+F)+5)/6) \times 272 \]

where \( S \) is the number of segments in a PCB, and \( F \) is the number of fields in a PCB.

After calculating the size for each PCB within a PSB, add the results to get the total storage size of the DXTPSB description.
Estimating storage for DXTVIEW descriptions
For each DXTVIEW description you create, the UIM creates two types of records in the FDTLIB:

- Internal records
- Source records

DataRefresher uses internal DXTVIEW records to decide which fields to use when processing extract requests. DataRefresher uses source view records to respond to a PRINT or PUNCH command. To estimate the storage needed for a DXTVIEW description, you must calculate the size for both internal and source records.

**Estimating storage for internal DXTVIEW descriptions:** Use the following formula to estimate the storage requirements in bytes for one internal DXTVIEW description:

\[ \text{number of bytes} = (F \times 154) + (((F+10)/11) \times 286) \]

where \( F \) is the number of fields visible to the view.

**Estimating storage for source DXTVIEW descriptions:** Use the following formula to estimate the storage requirements in bytes for one source DXTVIEW description:

\[ \text{number of bytes} = (F \times 96) + (((F+17)/18) \times 236) \]

where \( F \) is the number of fields explicitly listed in the FIELDS keyword of the CREATE DXTVIEW command.

Allocating FDTLIB storage
You need to define the FDTLIB as a VSAM key-sequenced data set (KSDS) so that it can store data descriptions.

To designate the size of an FDTLIB, you need to specify in your Access Method Services job:

- The type and amount of storage that the data set (or cluster) needs in the volume (TRACKS parameter).
- The average and maximum size of the records in the cluster (RECORDSIZE parameter).

Figure 23 on page 90 shows an example of the JCL and the Access Method Services job you use to create an FDTLIB. You can use this as a basis for your job, supplying values that are valid for your site.
Figure 23. JCL and Access Method Service job for creating an FDTLIB

1 nnnnnn is the ID of the volume to contain the cluster.
2 uuuu is the device type.
3 Communicate the name(s) of the FDTLIB(s) that you specify in the DEFINE CLUSTER statement of the Access Method Services job to the person who writes the JCL for running the DEM. The DEM needs these names to process extract requests that reference DataRefresher UIM/DEM data descriptions stored in the FDTLIB(s).

In this example, the name of the FDTLIB is user-prefix+FDTLIB.

4 nnnnnn is the ID of the volume to contain the cluster.
5 Designate the storage of a cluster using one of three types of units:

- TRACKS
- CYLINDERS
- RECORDS

In this example the FDTLIB records are stored in TRACKS. The numeric values that follow the parameter indicate the amount of storage you are assigning for that cluster which, in this example, is the number of tracks you allocate.

The first value (the primary amount) is the amount of storage initially allocated. The second value (the secondary amount) is the amount of storage allocated each time the data set needs to be extended. In this example, five tracks of storage were specified for both primary and secondary storage.

Consult your system programmer for information about the sizes for your device before using the results of your calculations to determine the amount of space needed in your FDTLIB because track, cylinder, and record sizes vary according to device.
Use the RECORDSIZE parameter to indicate the size of the records in the cluster. Two numeric values are associated with the parameter:

- The first value specifies the average record size (in bytes).
- The second value specifies the maximum record size (also in bytes).

In this example job, the average record size in the FDTLIB cluster is 1500 bytes and the maximum record size in the cluster is 2000 bytes.

Use the results of the calculations you made for your data descriptions to specify the average and maximum record sizes in the RECORDSIZE parameter.

For more information about defining VSAM clusters, see the appropriate VSAM Administration Guide or Integrated Catalog Facility Administration Guide. For more information about the individual job statements, see the appropriate Access Method Services Reference.

Reallocating FDTLIB storage

If you are not allocating an FDTLIB for the first time, but instead want to reallocate the size of the existing FDTLIB while keeping the present entries intact:

1. Create a new FDTLIB (using a new name) with the appropriate amount of space, using the Access Method Services DEFINE command.

2. Copy the existing FDTLIB entries into the new FDTLIB, using the Access Method Services REPRO command.

Figure 24 shows an example of the JCL and the Access Method Services job you use to copy one data set into another. You can use this as a basis for your job, supplying values that are valid for your site.

```
//FDTREP EXEC PGM=IDCAMS
/*********************COPYING FDTLIB*******************************
//INDSET1 DD DNAME=****user-prefix****.FDTLIB,DISP=OLD
//OUTDSET1 DD DNAME=****user-prefix****.FDTLIB1,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
REPRO -
   INFILE(INDSET1) -
   OUTFILE(OUTDSET1)
```

Figure 24. JCL and Access Method Services job for copying one data set into another

3. Delete the old FDTLIB, using the Access Method Services DELETE command.

Figure 25 on page 92 shows an example of the JCL and the Access Method Services job that you can use to delete a data set. You can use this as a basis for your job, supplying values that are valid for your site.
4. Rename the new FDTLIB, using the Access Method Services ALTER command.

After deleting the old data set, you may want to rename your newly created FDTLIB using the name of the old cluster. Figure 26 shows an example of the JCL and the Access Method Services job you use to rename a data set. You can use this as a basis for your job, supplying values that are valid for your site.

Administering the EXTLIB

The amount of space you allocate for an EXTLIB depends on several factors, for example:

- The more extract requests users submit, the more space the EXTLIB needs.
- The frequency with which the DEM is run influences the size of the EXTLIB. If the DEM is run in continuous-run mode, there are probably fewer extract requests waiting in the EXTLIB at any one time than if the DEM is run in terminating-run mode.

After you define your EXTLIB, you can increase or decrease its size depending on the changing requirements at your site.

Estimating EXTLIB storage

To create or reallocate an EXTLIB, you must estimate the size and number of requests waiting in the library at any one time. This section gives you both a space estimate for an average extract request and a formula to help you estimate the amount of space you need for the extract requests waiting for processing.

The storage for an extract request can vary from 1760 to over 1 million bytes. To estimate how much space you need in your EXTLIB for extract requests, you need to know such things as:
• The number of fields from which the request extracts data
• The number of WHERE conditions
• The number of lines of JCS to load the data into the database table

For example, an extract request that:

• Extracts from 25 fields
• Places data into 25 columns in the database target table
• Has three WHERE conditions
• Contains 25 lines of JCS to load the data into the database table
• Has a USERDECK string 12 bytes long

requires about 7400 bytes of storage. If your extract request contains long character constants in its WHERE clause, the request may require more space than other extract requests with different information in their WHERE clauses. But an average extract request uses about 8000 bytes of storage.

Use the following formula to estimate the storage requirements in bytes for one extract request:

$$\text{number of bytes} = 1760 + (104 \times S) + (21 \times C) + (160 \times Q) + (80 \times J) + U$$

where:

- **S** is the number of selected fields in the extract request.
- **C** is the number of columns in the target table (if there is an INTO clause in the extract request).
- **Q** is the number of WHERE clause qualifiers (if there is a WHERE clause in the extract request).
- **J** is the number of lines of JCS associated with the extract request. (Includes both extract output JCS and message JCS.)
- **U** is the length of the USERDECK string (if the USERDECK keyword was specified) excluding the delimiting single quotes.

**Allocating EXTLIB storage**

You need to define EXTLIB as a VSAM KSDS so it can store extract requests as they wait for processing.

To designate the size of an EXTLIB, you need to specify in your Access Method Services job:

• The type and amount of storage that the data set (or cluster) needs in the volume (TRACKS parameter)
• The average and maximum size of the records in the cluster (RECORDSIZE parameter)

Figure 27 on page 94 shows an example of the JCL and the Access Method Services job that you use to create an EXTLIB. You can use this as a basis for your job, supplying values that are valid for your site.
Figure 27. JCL and Access Method Services job for creating an EXTLIB

1. nnnnn is the ID of the volume to contain the cluster.

2. uuuu is the mass storage device type.

3. Communicate the name(s) of the EXTLIB(s) that you specify in the DEFINE CLUSTER statement of the Access Method Services job, to the person who writes the JCL for running the DEM and UIM. The UIM and DEM need the names to process extract requests stored in the EXTLIB(s).

   In this example, the name of the EXTLIB is user-prefix.EXTLIB.

4. nnnnn is the ID of the volume to contain the cluster.

5. Designate the storage of a cluster using one of three types of units:
   - TRACKS
   - CYLINDERS
   - RECORDS

   In this example the EXTLIB records are stored in TRACKS. The numeric values that follow the parameter indicate the amount of storage you are assigning for that cluster which, in this example, is the number of tracks you allocate.

   The first value (the primary amount) is the amount of storage initially allocated. The second value (the secondary amount) is the amount of storage allocated each time the data set needs to be extended. In this example, five tracks of storage were specified for both primary and secondary storage. Before using the results of your calculations to determine the amount of space needed in your EXTLIB, consult your system programmer for information about the sizes for your device. Track, cylinder, and record sizes vary according to device.
Use the RECORDSIZE parameter to indicate the size of the records in the cluster. Two numeric values are associated with the parameter:

- The first value specifies the average record size in bytes.
- The second value specifies the maximum record size in bytes.

In this example job, the average record size in the EXTLIB cluster is 1200 bytes and the maximum record size in the cluster is 4000 bytes. To specify your average and maximum record sizes, use the calculations you performed to estimate your EXTLIB storage requirements.

For more information about defining VSAM clusters, see the appropriate VSAM Administration Guide or Integrated Catalog Facility Administration Guide. For more information about the individual statements, see the appropriate Access Method Services Reference.

### Reallocating EXTLIB storage

If you are not allocating an EXTLIB for the first time, but want to reallocate the size of an existing library while keeping the present entries intact, do the following:

1. Create a new EXTLIB (using a new name) with the appropriate amount of space by using the Access Method Services DEFINE command.

2. Copy the existing EXTLIB entries into the new EXTLIB, using the Access Method Services REPRO command.

   Figure 28 shows an example of the JCL and the Access Method Services job that you use to copy one data set into another. You can use this as a basis for your job, supplying values that are valid for your site.

```
//EXTREP EXEC PGM=IDCAMS
//******************************************************************************
// COPYING EXTLIB
******************************************************************************
//INDSET1 DD DNAME=****user-prefix****.EXTLIB,DISP=OLD
//OUTSET1 DD DNAME=****user-prefix****.EXTLIB1,DISP=OLD
//SYSPRINT DD SYSSOUT=A
//SYSSIN DD *
//REPRO -
//INFILE(INDSET1) -
//OUTFILE(OUTSET1)
```

**Figure 28. JCL and Access Method Services job for copying one data set into another**

3. Delete the old EXTLIB, using the Access Method Services DELETE command.

   Figure 29 shows an example of the JCL and the Access Method Services job that you can use to delete a data set: You can use this as a basis for your job, supplying values that are valid for your site.

```
//EXTDEL EXEC PGM=IDCAMS
******************************************************************************
// DELETING EXTLIB
******************************************************************************
//INDSET1 DD DNAME=****user-prefix****.EXTLIB,DISP=OLD
//SYSPRINT DD SYSSOUT=A
//SYSSIN DD *
//DELETE -
//****user-prefix****.EXTLIB -
//CLUSTER
```

**Figure 29. JCL and Access Method Services job for deleting a data set**
4. Rename the new EXTLIB, using the Access Method Services ALTER command.

After you delete the old data set, you may want to use the name of the old cluster to rename your newly created EXTLIB. Figure 30 shows an example of the JCL and the Access Method Services job that you use to rename a data set. You can use this as a basis for your job, supplying values that are valid for your site.

```
//EXTREN EXEC PGM=IDCAMS
//**************************************************************************************
//* RENAMING EXTLIB
//**************************************************************************************
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
ALTER -
   ***user-prefix***.EXTLIB1 -
   NEWNAME(***user-prefix***.EXTLIB)
```

*Figure 30. JCL and Access Method Services job for renaming a data set*
Chapter 10. Setting up the Data Extract Manager (DEM)

DataRefresher provides you with the capability to simplify the scheduling and processing of extract requests to suit your site's operation priorities. You can set up the DEM to operate in specific ways to meet your data handling requirements. Preparing for DEM operation involves:

- Setting up the DEM environment
- Specifying source and output data sets for your extract requests
- Batching extract requests (optional)

This chapter provides information on setting up the DEM at your site.

- "Deciding what type of DEM to run" provides guidelines to help you decide what type of DEM will be consistent with your site's operation priorities.

- "Setting up the DEM environment" on page 98 provides guidelines for setting a DEM's run mode, priority range, dynamic data set allocation defaults, output limits, and diagnostic output requirements.

- "Determining which extract requests a DEM processes" on page 102 describes how the DEM determines which extract requests to process.

- "Information necessary to write DEM JCL and initialization commands" on page 102 describes what information you need to run different types of DEMs, and who can supply you with that information.

- "Specifying source data sets for DEM extract requests" on page 103 provides information that you need to access source data sets when running non-IMS batch jobs, IMS batch jobs, and IMS BMP jobs.

- "Specifying output data sets for DEM extract requests" on page 107 describes how to specify output data sets to the DEM.

- "Batching extract requests" on page 108 describes how you can batch extract requests so that the DEM can run more than one extract request at the same time.

- "Setting up a message table" on page 111 describes the table you can create to receive messages regarding the status of extract requests.

Deciding what type of DEM to run

You can run the DEM as one of the following:

- Online in the TSO foreground
- Non-IMS MVS batch job
- IMS batch job
- IMS batch message processing (BMP) job
- CICS batch job

Table 13 on page 98 shows the relationship between the type of data a DEM extracts and the type of DEM you should run.
Table 13. Types of DEMs suitable for types of data

<table>
<thead>
<tr>
<th>If you have this type of data</th>
<th>Run this type of DEM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Online DEM in TSO</td>
</tr>
<tr>
<td></td>
<td>foreground</td>
</tr>
<tr>
<td></td>
<td>Non-IMS MVS batch job</td>
</tr>
<tr>
<td></td>
<td>IMS batch job</td>
</tr>
<tr>
<td></td>
<td>IMS BMP job</td>
</tr>
<tr>
<td></td>
<td>CICS batch job</td>
</tr>
<tr>
<td>IMS batch databases</td>
<td>X</td>
</tr>
<tr>
<td>IMS DL/I online databases</td>
<td>X</td>
</tr>
<tr>
<td>VSAM data sets</td>
<td>X</td>
</tr>
<tr>
<td>Physical sequential data sets</td>
<td>X</td>
</tr>
<tr>
<td>GDI data sources</td>
<td>X</td>
</tr>
</tbody>
</table>

While all these ways of running a DEM give a DEM access to VSAM data, physical sequential data, and GDI data, no one way of running a DEM gives a single DEM access to all the types of data. If you have all of these types of data to process at your site, you must create at least two DEMs to access it all. However, you might want to organize data by creating the following DEMs:

- Online DEM in the TSO foreground to process small extract requests for VSAM and physical sequential data sets
- Non-IMS MVS batch job to access VSAM and physical sequential data sets
- IMS batch job to access just IMS DL/I batch databases
- IMS BMP job to access just IMS DL/I online databases

For information on running single or multiple DEMs, see “Preparing for Data Extract Manager operation” on page 82.

Setting up the DEM environment

You use the INITDEM command to set up the DEM operating environment. This command identifies a DEM and specifies a DEM’s characteristics, which include:

- Run mode and polling interval (RUNMODE and POLLINTV)
- Priority range (PRILIM)
- Defaults for dynamic data set allocation (DYNAMOUT)
- Output limit for extract requests (OUTLIM)
- Diagnostic output requirements (DEBUG)

Setting the run mode and polling interval for a DEM

When scheduling a DEM, you must decide whether to run it in either:

- Continuous-run mode
- Terminating-run mode

**Continuous-run mode:** If you run a DEM in continuous-run mode, it runs all qualifying extract requests in the EXTLIB, and then periodically rechecks the EXTLIB for new extract requests to run (thus making it appear to DataRefresher users that the DEM runs simultaneously with the UIM). However, when you run a DEM in continuous-run mode, you can only do so for the period you specify. At the end of that period, the DEM changes to terminating-run mode.

The polling interval is the period that a DEM running in continuous-run mode waits, after running all appropriate extract requests in the EXTLIB, before rechecking it for new extract requests to run. When deciding the length of a polling interval, realize
that the shorter you make the polling interval, the more interactive the DEM and UIM will appear to DataRefresher users. A shorter polling interval is also more likely to cause the DEM to spend time looking in an empty EXTLIB for new extract requests to run and unnecessarily using up system resources.

**Terminating-run mode:** If you run a DEM in terminating-run mode, it runs all qualifying extract requests in the EXTLIB, and then stops.

Suppose, for example, that you want to create two DEMs:

- One DEM to run high-priority extract requests as quickly as possible during peak hours when DataRefresher users submit them
- One DEM to run all remaining extract requests during off-peak hours

You should schedule the first DEM to run during peak hours in continuous-run mode, and the second to run during off-peak hours in terminating-run mode. When you run the first DEM, it processes all high-priority extract requests in the EXTLIB, and then periodically rechecks the EXTLIB for new high-priority requests to process.

When you run the second DEM, it processes all low-priority extract requests, and then stops. However, if DataRefresher users continue to submit extract requests that meet the DEM's requirements for running extract requests, the second DEM (if it has not yet terminated) processes the newly submitted extract requests. If you run a DEM in terminating-run mode, it does not stop until it checks the EXTLIB for extract requests that it can run and finds none.

To establish whether a DEM runs in terminating or continuous-run mode, and, if the latter, to establish a polling interval, you must specify the necessary information on the INITDEM command.

**Note:** An MVS operator who controls a DEM can change the DEM’s run mode and polling interval. See the DataRefresher MVS and VM User’s Guide for an explanation of how, and under what circumstances, this should be done.

### Setting up extract request priorities

You can set up a DEM to run only those extract requests with a specified priority. For example, to define a priority range, choose numbers from 0 through 255 to represent levels or classes of priority within a given range.

Table 14 shows a hypothetical organization of priority ranges and their corresponding priority classes.

<table>
<thead>
<tr>
<th>Priority range</th>
<th>Priority class</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 - 255</td>
<td>Special authorization requests</td>
</tr>
<tr>
<td>150 - 199</td>
<td>Other priority requests</td>
</tr>
<tr>
<td>100 - 149</td>
<td>Low priority requests, process during off-peak hours</td>
</tr>
<tr>
<td>50 - 99</td>
<td>Low priority requests, process over the weekend</td>
</tr>
<tr>
<td>0 - 49</td>
<td>Reserved for testing</td>
</tr>
</tbody>
</table>
If you want to run some extract requests as quickly as possible during peak hours, but want to run most extract requests during off-peak hours, create two DEMs:

- One DEM to run high-priority extract requests as quickly as possible during peak hours when DataRefresher users submit them
- One DEM to run all remaining extract requests during off-peak hours

Based on the priority ranges given in Table 14 on page 99, only those high-priority extract requests with a priority range of 200 through 255 will run during the day.

You establish a priority range by specifying the PRILIM keyword in the INITDEM command. A single range of priority numbers is assigned to each run of the DEM through the INITDEM command. These priority ranges provide numeric bounds to define the limits of a priority class. Thus, DataRefresher users can assign a priority value to each of their extract requests (the higher the value the higher its priority) and a DEM can be set up to run only those extract requests having a priority value falling within the designated priority range.

DEM also tries to run the extract requests falling within its priority range in order of priority but other factors such as batching extract requests can influence which extract requests the DEM will run. For example, the fact that an extract request is batched together with multiple extract requests is primary to its priority value. So, the DEM will run those extract requests that are batched before single extract requests that have a high priority value.

The DEM operator can change a DEM's priority range, and can use the LIST command to decide when to change the range. See the DataRefresher MVS and VM User's Guide for an explanation of how, and under what circumstances, this should be done.

**Setting defaults for dynamic allocation**

You can specify the defaults for dynamic allocation of physical sequential data sets in the INITDEM command. You can override these defaults by specifying the EXTDATA keyword in the SUBMIT command.

Using the DataRefresher INITDEM command, you can specify the volume of output associated with dynamic allocation that goes to the physical sequential data set.

The physical sequential data set output from an extract request can be dynamically allocated by a DEM with user-specified information. This user-specified information is included in the extract request when it is submitted. Where information is not specified, the INITDEM default values are used. You can override any or all of the values you might code for dynamic allocation, which include the:

- Unit of the DASD device
- Volume serial
- Number of extract records per block
- Maximum number of blocks

You override the INITDEM defaults by coding the EXTDATA keyword on the SUBMIT command. The EXTDATA keyword identifies a data set that the DEM dynamically allocates. The DEM then writes the extracted data to this data set, as shown in Figure 31 on page 101.
Setting up DEM output limits

You can set up a DEM to run only those extract requests that do not generate more than a specified number of output rows when the DEM runs. You specify the DEM's output limits with the OUTLIM keyword on the INITDEM command. You can choose to set up two DEMs, for example, to run:

- Small output extract requests during the same day DataRefresher users submit them
- Large output extract requests during off-peak hours

DataRefresher users specify how many output rows a DEM should write per extract request, and must code an output limit within a DEM's DXTIN data set. Thus, DataRefresher users can include in each extract request an estimate of the number of output rows it writes when processed. A DEM runs only those extract requests that have an output row estimate that is less than or equal to the DEM's output limit. Because the DEM stops processing an extract request when the specified output estimate is reached, an incomplete output will result from an estimate that is too low.

The DEM operator can change a DEM's output limit. See the DataRefresher MVS and VM User's Guide for an explanation of how, and under what circumstances, this should be done.

Determining diagnostic output requirements

Each DEM can provide diagnostic information to help you resolve any problems that occur while it is running. You control the amount of diagnostic information that you get from running a DEM by coding a debug level (level of diagnostic information) on the DEBUG keyword in the INITDEM command. Refer to DataRefresher Command Reference for a list of the debug levels you can specify.

The higher the debug level you request, the more diagnostic information a DEM outputs. However, the higher the debug level, the more the performance of the DEM will be impacted. Use debug level 1 (the default value) unless there are specific problems requiring more detailed diagnostic information.

For some system errors, the system prints control area and trace information regardless of what debug level you choose.
A DataRefresher user can assign a debug level to a given extract request. Such a debug level applies only if it is higher than the debug level you establish for the DEM that runs that extract request, and only while that DEM runs it. A DEM can also batch extract requests to be run together at the same time. (See “Batching extract requests” on page 108 for details.) However, if a DataRefresher user assigns a debug level higher than 1 to a given extract request, the DEM that runs that extract request does not batch it with any other extract requests.

The DEM operator can change a DEM’s debug level. See the DataRefresher MVS and VM User’s Guide for an explanation of how, and under what circumstances, this should be done.

**Determining which extract requests a DEM processes**

A DEM can run an extract request only if all of the following occur:

- The initialization commands let a DEM access the requested data
- The request's priority value falls within a DEM's priority range
- The request's output row estimate is less than or equal to a DEM's output limit
- The IDs on the extract requests match (if specified)

To determine which extract request to process:

1. DataRefresher tries to match extract request IDs between the extract request (EXTID keyword) and the USE EXTNID statement in the DEM JCL. If these IDs do not match, DataRefresher will not process the extract request.

2. If the extract request IDs match, DataRefresher reads the USE DXTFILE or USE DXTPSB statement.

3. DataRefresher checks to see whether the extract request meets the DEM environment's parameters. If so, the extract request is run. If not, the extract request remains in the EXTLIB to be processed.

You can use the LIST command to help determine the characteristics of the extract requests in the EXTLIB. For example, you can determine what priorities the extract requests have and what files or PSBs the extract requests use. This information can help you set up the DEM initialization commands so that the DEM processes a given set of extract requests.

**Information necessary to write DEM JCL and initialization commands**

Table 15 shows the type of information you need to run different types of DEMs and who can supply you with that information. Note that the DataRefresher administrator and the database administrator may be the same person.

<table>
<thead>
<tr>
<th>Type of DEM</th>
<th>Necessary Information to write DEM JCL and initialization commands</th>
<th>Person with information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online DEM in the TSO foreground</td>
<td>Extract request ID</td>
<td>DataRefresher administrator</td>
</tr>
<tr>
<td></td>
<td>DD names</td>
<td>DataRefresher administrator</td>
</tr>
<tr>
<td></td>
<td>Data set name</td>
<td>Owner of the data set</td>
</tr>
</tbody>
</table>
### Table 15 (Page 2 of 2). Where to find information for running types of DEMs

<table>
<thead>
<tr>
<th>Type of DEM</th>
<th>Necessary information to write DEM JCL and initialization commands</th>
<th>Person with information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-IMS MVS batch job</td>
<td>DXTFILE description name</td>
<td>DataRefresher administrator</td>
</tr>
<tr>
<td></td>
<td>Extract request ID</td>
<td>DataRefresher administrator</td>
</tr>
<tr>
<td></td>
<td>DD names (if the ddname is taken from the DXTFILE description)</td>
<td>DataRefresher administrator</td>
</tr>
<tr>
<td></td>
<td>Data set name</td>
<td>Owner of the data set</td>
</tr>
<tr>
<td>IMS batch job</td>
<td>IMS PSB name</td>
<td>Database administrator</td>
</tr>
<tr>
<td></td>
<td>DXTPSB and DXTPCB description names</td>
<td>DataRefresher administrator</td>
</tr>
<tr>
<td></td>
<td>Extract request ID</td>
<td>DataRefresher administrator</td>
</tr>
<tr>
<td></td>
<td>DD names for databases (from DBDs for the referenced database)</td>
<td>Database administrator</td>
</tr>
<tr>
<td></td>
<td>Data set names for databases</td>
<td>Database administrator</td>
</tr>
<tr>
<td>IMS BMP job</td>
<td>IMS PSB name</td>
<td>Database administrator</td>
</tr>
<tr>
<td></td>
<td>DXTPSB and DXTPCB description names</td>
<td>DataRefresher administrator</td>
</tr>
<tr>
<td></td>
<td>Extract request ID</td>
<td>DataRefresher administrator</td>
</tr>
</tbody>
</table>

As the DataRefresher administrator, you should work with an operations planner to determine what data a given DEM can access. This is not solely the task of the DEM operator.

### Specifying source data sets for DEM extract requests

The figures in the following sections show additional information that you need to access source data sets when running the following types of DEMs:

- Non-IMS MVS batch job
- IMS batch job
- IMS BMP job

**Non-IMS MVS batch job**

The following figures summarize the information that you need to run a non-IMS MVS batch job, and show ways of identifying where the data you want to extract resides using the:

- USE EXTID statement
- USE DXTFILE statement
- CREATE DXTFILE command
USE EXTTID statement
Figure 32 shows how the USE EXTTID statement identifies the extract request that specifies the DXTVIEW from which to extract data:

1 The DEM uses dxfile to find the data description in the FDTLIB.
2 The USE DXTFILE statement does not contain a ddname operand that points to a DD statement. The DDNAME keyword in the CREATE DXTFILE command names the data set from which data will be extracted.
3 The DSN DD statement in the DEM JCL identifies the data set containing the data to be extracted.
4 The DEM uses ext1 to find the extract request in the EXTLIB.
5 The DXTVIEW used by the extract request (ext1) gets data from the named dxfile.

USE DXTFILE statement
Figure 33 shows how the USE DXTFILE statement in the DEM JCL overrides the DD statement in the CREATE DXTFILE command:

1 The DEM uses dxfile to find the data description in the FDTLIB.
2 The USE DXTFILE statement contains a ddname operand that points to a DD statement. The DD statement names the data set from which data will be extracted, and it overrides the DDNAME keyword in the CREATE DXTFILE command.

3 The DSN DD statement in the DEM JCL identifies the data set containing the data to be extracted.

CREATE DXTFILE command

Figure 34 shows how the DD statement in the CREATE DXTFILE command identifies the data set from which you want to extract data:

1 The DEM uses dxfile to find the data description in the FDTLIB.

2 The USE DXTFILE statement does not contain a ddname operand that points to a DD statement. The DDNAME keyword in the CREATE DXTFILE command names the data set from which data will be extracted.

3 The DSN DD statement in the DEM JCL identifies the data set containing the data to be extracted.
IMS batch job

Figure 35 summarizes the information you need to run an IMS batch job:

1. The **imspsb** in the EXEC statement points to the name of the IMS PSB in the IMS PSBLIB. The DEM will use this PSB to access IMS DL/I data.

2. The USE DXTPSB statement references the data description of the DXTPSB in the FDTLIB.

3. The PCB statements point to DBD statements in the IMS DBDLIB. The names of IMS DBDs, stored in the IMS DBDLIB, define IMS databases from which data will be extracted.

4. The DBD statements contain DD names for IMS databases from which data will be extracted.

5. **dsn1** and **dsn2** are the names of data sets in the DEM JCL that comprise the IMS databases.
Figure 36 summarizes the information you need to run an IMS BMP job:

1. The **imspsb** in the EXEC statement points to the name of the IMS PSB in the IMS ACCLIB. The DEM will use this PSB to access IMS DL/I data.

2. The USE DXTPSB statement references the data description of the DXTPSB in the FDTLIB.

3. DBD statements containing DD names are in the ACCLIB. The ddnames describe IMS databases from which data will be extracted.

4. **dsn1** and **dsn2** in the DD statements are data set names including the IMS databases allocated to the IMS Online Control Region.

### Specifying output data sets for DEM extract requests

You can specify an output data set to the DEM by:

- Specifying a Generic Output Interface (GOI) exit routine by naming the exit routine on the GOEXIT keyword on the SUBMIT command
- Specifying the EXTDATA keyword on the SUBMIT command
- Directing the extracted data to an internal reader
- Dynamically allocating a data set

### Specifying a GOI exit routine using the GOEXIT keyword

By using a GOI exit routine, you can transform your data into a user-defined format and direct it to a target otherwise unsupported by DataRefresher. You specify your exit routine on the GOEXIT keyword of the SUBMIT command. For information on writing GOI exit routines, see the *DataRefresher Exit Routines*. 

---

Chapter 10. Setting up the Data Extract Manager (DEM) 107
Allocating a data set using the EXTDATA keyword

Figure 37 shows how to allocate a data set for physical sequential output using the EXTDATA keyword:

1. The extract request that the DEM selects for processing names a DD statement in the DEM JCL.
2. The data set name (dsn) in the DEM JCL refers to a data set to which the DEM will write extracted data.

Directing extracted data to an internal reader

You can tell DataRefresher to put extracted data in your reader by specifying the JCS keyword in your extract request and a DXTOUT1 DD statement in your DEM JCL. The DEM will write at least the control deck to the DXTOUT1 DD statement which can be defined as the internal reader (for example, //DXTOUT1 DD SYSOUT=(A, INTRDR)).

Dynamically allocating a data set

The physical sequential data set output from an extract request can be dynamically allocated by a DEM with user-specified information. This user-specified information is included in the extract request when it is submitted. For more information, see “Setting defaults for dynamic allocation” on page 100.

Batching extract requests

The DEM can run more than one extract request at a time during a single pass of the source database or file. This process of batching extract requests, that is, running extracts together, can save a considerable amount of time and processing resources over running requests one at a time.

The number of extract requests which can be batched is limited by the number of DDname=DXTOUTn data sets allocated to the DEM, where the value of n can range from 1 to 99 inclusive. For example, if the DEM job contains DD statements with DD names DXTOUT1 through DXTOUT10, up to ten extract requests can be batched (run together) during the same run of the DEM. Note that:

- Dynamic allocation of a data set for physical sequential output includes part of the total 99 DXTOUTn data sets that can be batched together.
• DEM DXTOUTn data sets can be sequential data sets, JES internal readers, or other output devices. Generally, you specify your DEM DXTOUTn data sets as JES internal readers.

• You should not mix DXTOUTn JES internal readers and sequential data sets. (This is to ensure that the output of certain requests go to JES internal readers when necessary. DataRefresher cannot distinguish between JES internal readers and other DXTOUT devices.)

• A DEM's DXTOUTn data sets are established through its JCL job stream.

• The DEM operator cannot control which extract requests the DEM batches together.

Batching high-priority extract requests
Remember that there is always a high-priority extract request which runs first. The high-priority request is the one with the highest priority within the DEM's priorities (defined by the INITDEm command) addressing the DEM's source data (defined by USE commands). Among extract requests with equal priority, the oldest request has the highest priority.

If this high-priority extract request can be batched (see "Determining which extract requests to batch"), and the batching number (the n of DXTOUTn) is greater than one, then the DEM looks for extract requests which can be run with the high-priority request, even though these extract requests may have been submitted at a lower priority.

Determining which extract requests to batch
DataRefresher can batch an extract request (except for extract requests containing an ORDER BY clause, or a GDI Select exit routine) with other requests if all of the following criteria are met:

• The job to run the DEM must contain multiple DXTOUTn DD statements. An insufficient number of DXTOUTn DD statements will cause DataRefresher to make at least two passes of the database or file.

• Each request must have a priority number which matches those priority numbers that the DEM is set up to run.

• Each request extracts data from a single source file or database. This means that the FROM clause must contain either:
  - The name of a single DXTVIEW
  - The names of multiple DXTVIEWs that are all based on the same DXTPCB (and thus on the same IMS database)

• The OPTIONS clause of the EXTRACT statement in each request does not contain a DEBUG value greater than 1. To produce additional diagnostic information for a batch of extract requests, specify DEBUG=n on the INITDEm command, where n is a number from 1 to 4.

• The requests do not extract data from source data sets and databases which support direct access (VSAM KSDS, and the IMS DL/I HDAM, HISAM or HIDAM organizations), and which use a WHERE clause such as:
  WHERE keyfield = value
  or
  WHERE keyfield IN (value1, value2, value3)
Batching extract requests such as these uses more resources than running the extract requests alone because DataRefresher can directly retrieve those records.

- The requests do not extract data from source data sets and databases that require an end-of-data call to a data exit, and that use a WHERE clause.

Certain types of data sources do not batch well. However, if extract requests with the same source file or database meet the batching criteria described in the previous discussion, and address overlapping ranges of the data set or database, the requests are run together. (Requests may address overlapping ranges as determined by specifications made on the WHERE clauses, referencing key field or sequence fields.)

Note the following:

- All extract requests running against physical sequential data sets, VSAM data sets, GDI data sources, HSAM IMS DL/I databases, or HDAM IMS DL/I databases, that can be batched together, can be run with other such requests.

- Extract requests for VSAM KSDS data sets and HISAM and HIDAM databases that can be batched together, and that address overlapping ranges, are actually run together.

The example in Figure 38 shows a keyed data set (FILEA) and three extract requests that address certain portions of the keyed data set.

![Figure 38. Three extract requests which address a keyed data set](image)

In this example, Extract Request B and Extract Request C would be run together because there are some records that are common to both extract requests. Extract Request A would be run separately because it needs records that do not overlap with the other two extract requests.

However, if the WHERE clause for Extract Request A was specified as:

WHERE keyfield BETWEEN 000100 AND 200200

then all three extract requests would be run together.

- Extract requests written without WHERE clause qualifiers (against the database key) overlap all other extract requests that can be batched together for that
data set or database. Thus, these extract requests are batched together with other requests.

**Note:** If DataRefresher is directed to read an entire unordered data set or database, DataRefresher will read all the records and test each one. Here, DataRefresher will batch together extract requests with dissimilar keys.

Extract requests that traverse the same hierarchical paths within an IMS DL/I database are often batched together for that access. For example, Extract Request 1, Extract Request 2, and Extract Request 3 are run against the following database hierarchy:

```
    A
     |
    -- B
     |
     C
```

- Extract Request 1 extracts from A and C.
- Extract Request 2 extracts from A and B.
- Extract Request 3 extracts from A and C.

If the three extract requests are batched at the root level, Extract Request 1 and Extract Request 3 could share an access to segment C, and Extract Request 2 would access segment B alone.

---

### Setting up a message table

Messages regarding extract requests, STATUS commands, and CANCEL commands issued during a DEM run can be received into a message table, that is, a relational database table set up to receive DataRefresher messages. To do this, create a table with columns such as those described in Table 16. This message table can help you check the results of extract requests you run during off-peak hours.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Column description</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>USERID</td>
<td>Contains the value of the USERID keyword of the extract request, STATUS command, or CANCEL command to which the message applies.</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>EXTDID</td>
<td>Contains the value of the EXTDID keyword of the extract request, STATUS command, or CANCEL command to which the message applies.</td>
<td>CHAR(18)</td>
</tr>
<tr>
<td>TIME</td>
<td>Contains the time at which the message was sent. The time has the following format: hh:mm:ss, where hh are the hours, mm are the minutes, and ss are the seconds.</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>DATE</td>
<td>Contains the date on which the message was sent. The date has the following format: mm/dd/yy, where mm is the month, dd is the day, and yy is the year.</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>EVENT</td>
<td>Contains a three-digit event code of 001 through 255. An event is the situation that resulted in a message being issued. All messages about the same event will be identified with the same event code. After an event code of 255 is issued, the next event code will be 001.</td>
<td>CHAR(3)</td>
</tr>
</tbody>
</table>

---

Chapter 10. Setting up the Data Extract Manager (DEM) 111
### Table 16 (Page 2 of 2). DEM message table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Column description</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSGNUM</td>
<td>Contains the message number.</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>APPEND</td>
<td>Contains DataRefresher operator messages. These messages are not sent to DataRefresher users.</td>
<td>CHAR(1)</td>
</tr>
<tr>
<td>MSGTEXT</td>
<td>Contains the message text.</td>
<td>LONG VARCHAR</td>
</tr>
</tbody>
</table>

For information on how to create a DB2 table, see *IBM DATABASE 2 Version 2: SQL Reference*.

For information on how to create an SQL/DS table, see *SQL/Data System Application Programming for IBM VM Systems*.

## Administering DEM

This section describes how and under what circumstances an MVS operator should intervene in the running of a DEM. It also discusses what to do when a DEM ends abnormally.

MVS operator tasks to control the DEM include:

- Communicating with the DEM
- Displaying DEM information
- Stopping a DEM
- Changing a DEM's operating characteristics
- Recovering from errors

## Communicating with a DEM

An MVS operator intervenes in the running of a DEM by communicating with that DEM through the MVS operator console chosen for controlling it.

Suppose, for example, that a DEM scheduled to run overnight in terminating run mode continues to run longer than expected. The MVS operator controlling that DEM first wants to find out how many extract requests that qualify for execution by that DEM are still in the EXTLIB. To do so, the MVS operator enters the appropriate DEM operator statement through the controlling MVS console.

Assume your overnight processing schedule allows time for the DEM to execute some but not all of the extract requests still contained in the EXTLIB. The MVS operator can then enter a DEM operator statement to change a DEM's priority range so that the DEM only executes any remaining high-priority extract requests. If, however, your overnight processing schedule allows no more time for executing extract requests, the MVS operator can enter a DEM operator statement that stops the DEM once it finishes executing its current batch of extract requests.

To actually communicate with a DEM, an MVS operator enters responses to invitational write-to-operator-with-reply (WTOR) console messages. DEM invitational WTORs take the form:

```
nn DVRMOIE DXT DATA EXTRACT MANAGER name IS READY TO ACCEPT OPERATOR COMMANDS```

112 Administration Guide
Where:

- *nn* is a WTOR response number assigned by the system and entered by an MVS operator when responding.
- *DVRM01E* is the invitational WTOR's message number.
- *name* identifies the DEM to which the WTOR and operator response apply. *name* refers to the same name you specify on the INITDEM command with the NAME= keyword.

An operator response takes the following form:

```
R nn, statement
```

Where

- *R* shows this is a response to a WTOR.
- *nn* must be replaced with the relevant WTOR response number.
- *statement* must be replaced with a DEM operator command (DISPLAY, CONDHALT, TERMINATE, or CHANGE).

After the MVS operator enters a response to an invitational WTOR message, and after completion of the operator command, the MVS operator console again displays a WTOR of the following form:

```
nn DVR5121I DXT DATA EXTRACT MANAGER name IS READY TO ACCEPT OPERATOR COMMANDS
```

**Note:** When you code the INITDEM command, you have the option of suppressing the invitational WTOR messages with the INVWTOR keyword. If INVWTOR=N has been coded on the INITDEM command, the invitational WTOR messages do not appear. When INVWTOR=N has been coded, the operator cannot enter commands to that DEM.

### Displaying DEM information

The MVS operator who controls a DEM can review some of that DEM's operating characteristics (those that do not help establish what data that DEM can access), and information about extract requests in the execution cycle. To do so, the MVS operator uses the DISPLAY command.

Suppose, for example, you have two DEMs running in continuous run mode at the same time (one named DEMONE having a short polling interval and executing high-priority extract requests, the other named DEMTWO having a long polling interval and executing all other extract requests). Suppose too that the originators of the high-priority extract requests complain that DEMONE does not complete their extract requests fast enough.

Now, rather than immediately shortening DEMONE's polling interval (discussed later in this appendix), the MVS operator controlling the DEMs can check to see whether DEMTWO has a light load at this time.
Assume DEMTWO’s last invitational WTOR says:

10 DVRMOIE DXT DATA EXTRACT MANAGER DEMTWO IS READY TO ACCEPT OPERATOR COMMANDS

The MVS operator can display the relevant information by entering the DISPLAY command and its COUNTS keyword as a WTOR response:

R 10, DISPLAY COUNTS

Noting that DEMTWO has a light load, the MVS operator can broaden its priority range (assuming both DEMs can access the same data and that their output limits are compatible) and shorten its polling interval so that it helps DEMONE to quickly execute some of the high-priority extract requests. (How to do so is discussed later in this appendix.) If, in such a case, you want DEMTWO’s polling interval to equal DEMONE’s polling interval, the MVS operator can verify DEMONE’s polling interval before changing DEMTWO’s polling interval. To do so, the MVS operator uses the DISPLAY command’s RUNPARMS keyword.

Assume DEMONE’s last invitational WTOR says:

04 DVRMOIE DXT DATA EXTRACT MANAGER DEMONE IS READY TO ACCEPT OPERATOR COMMANDS

The MVS operator would enter:

R 04, DISPLAY RUNPARMS

Suppose, finally, that DEMONE seems to be taking too long to execute a given batch of extract requests. The MVS operator can use the DISPLAY command’s STATUS keyword to find out what time DEMONE started executing its current batch of extract requests and what extract requests are in the batch. The MVS operator would enter:

R 04, DISPLAY STATUS

If the STATUS display does in fact show that DEMONE is taking too long to execute its current batch of extract requests, the MVS operator might want to stop DEMONE.

An alternative is to stop the batch processing. (See “Stopping a DEM” on page 115.)
Note: An MVS operator can enter more than one DISPLAY command at a time. Assume, for example, an MVS operator wanted to see DEMONE's COUNTS, STATUS, and RUNPARMS displays at the same time, and DEMONE's last invitational WTOR says:

13 DVRMO1E DXT DATA EXTRACT MANAGER DEMONE IS READY TO ACCEPT OPERATOR COMMANDS

The MVS operator would enter:

R 13, DISPLAY COUNTS,STATUS,RUNPARMS

Stopping a DEM

The MVS operator who controls a DEM can stop that DEM before it ends normally by using the TERMINATE command.

Suppose, for example, that an MVS operator decides that DEMONE is taking too long to execute its current batch of extract requests and wants to stop it immediately. The MVS operator uses the TERMINATE command's FAST keyword. Assume DEMONE's last invitational WTOR says:

14 DVRMO1E DXT DATA EXTRACT MANAGER DEMONE IS READY TO ACCEPT OPERATOR COMMANDS

The MVS operator would enter:

R 14, TERMINATE STOPMODE=FAST

If the operator enters this command, the extract requests are queued again in the EXTLIB. They become eligible for execution by any DEM for which those extract requests qualify.

As another example, suppose that a DEM is not taking too long to execute its current batch of extract requests, but it has started to exceed its normally allotted time and you cannot let it continue to run. In this case, the MVS operator would use the TERMINATE command's NORMAL keyword. Assume DEMONE's last invitational WTOR says:

14 DVRMO1E DXT DATA EXTRACT MANAGER DEMONE IS READY TO ACCEPT OPERATOR COMMANDS

The MVS operator would enter:

R 14, TERMINATE STOPMODE=NORMAL

Entering this command stops DEMONE — but only after DEMONE has finished executing its current batch of extract requests.
There may be times, such as when a user exit is in an infinite loop, that the MVS operator must use the MVS CANCEL command to stop the DEM. To clean up the extract request that was being processed, the UIM CANCEL command can be issued. This signals the DEM, on a future invocation, to delete that extract request from the EXTLIB.

**Stopping a DEM temporarily**
The MVS operator who controls a DEM can temporarily stop that DEM with the CONDHALT command.

Suppose, again, that DEMONE seems to be taking too long to execute a given batch of extract requests, but the MVS operator wants to let it run while reviewing DISPLAY STATUS information. The MVS operator would, in this case, use the CONDHALT command.

Assume DEMONE's last invitational WTOR says:

```
15 DVM01E DXT DATA EXTRACT MANAGER DEMONE IS READY TO ACCEPT
OPERATOR COMMANDS
```

The MVS operator would enter:

```
R 15, CONDHALT
```

This command temporarily stops DEMONE and displays what time DEMONE started executing its current batch of extract requests and what extract requests are in the batch. A prompt is also displayed, requesting the MVS operator to either use the RESUME command to restart DEMONE or use the HALTBATCH command to stop execution of the current batch of extract requests.

If the MVS operator does not respond within a certain time period, the prompt is repeated. If the MVS operator does not respond to the third appearance of the prompt, the DEM resumes execution.

If, after looking at the CONDHALT display, the MVS operator decides that DEMONE is not taking too long to execute its current batch of extract requests, the MVS operator restarts DEMONE by responding to the CONDHALT prompt with:

```
R 15, RESUME
```

If the MVS operator decides that DEMONE is taking too long to execute its current batch of extract requests (because, for example, it seems to be in an infinite loop), the MVS operator would stop DEMONE's execution of its current batch of extract requests by responding to the CONDHALT prompt with:

```
R 15, HALTBATCH
```

and then call a system programmer.
If an operator uses the HALTBATCH command, the extract requests are requeued in the EXTLIB. They become eligible for complete execution by any DEM for which the extract requests qualify.

**Changing a DEM's operating characteristics**

An MVS operator who controls a DEM can use the CHANGE command to change that DEM's run mode, polling interval, priority range, output limit, debug level, and the console to which operator messages are sent. Such changes last only for the duration of the DEM's current execution or until the MVS operator again changes the DEM's relevant characteristic(s). This allows an MVS operator to adjust your DEMs (while they execute) to meet any unusual conditions or demands that arise.

Suppose, again, that a DEM scheduled to run overnight in terminating run mode continues to do so longer than expected, and your overnight processing schedule allows time for the DEM to execute some but not all of the extract requests still contained in the EXTLIB. An MVS operator has two ways of limiting the extract requests that the DEM can execute. As previously shown, the MVS operator can narrow the DEM's priority range so that it can only execute high-priority extract requests. Or, the MVS operator can change the DEM's output limit so that it executes only those extract requests having small output row estimates.

Next, suppose you have set up a DEM to run in continuous run mode for eight hours and you normally run this DEM every day. If the originators of the extract requests complain that the DEM does not complete their extract requests quickly enough, an MVS operator can shorten its polling interval. Or, if, during your site's end-of-month processing period, you cannot afford to let the DEM run in continuous run mode all day long and you do not want to code the JCL for a special once-a-month DEM, an MVS operator can submit the DEM's normal JCL to start the DEM during a slow part of the day but then, after the schedule gets heavy again, change it to run in terminating run mode.

The ability to change a DEM's debug level or the console to which operator messages are sent provides extra operating flexibility. If, for example, an MVS operator feels that a DEM is executing extract requests and that processing has not completely halted, that operator can let the DEM continue to run while changing its debug level so that it prints more diagnostic information for a system programmer. Or, if an MVS operator must do other tasks at various locations in your facility while controlling DEMs, that operator can switch the routing code by which the DEM sends messages to an MVS console to always control the DEMs from the most conveniently located console.

Before using the CHANGE command, you can use the LIST command to get information about the extract requests in EXTLIB to determine whether you want to change a DEM's operating characteristics.

The following sections show examples of the CHANGE command. Each example discusses only one of the keywords of the CHANGE command, but in any CHANGE command you may include one, some, or all of its keywords.
Changing a DEM's run mode
To change a DEM's run mode or change how long a DEM operates in continuous run mode, an MVS operator uses the CHANGE command's RUNMODE keyword.

Assume you want an MVS operator to change the run mode of a DEM named DEMONE from continuous to terminating run mode and that DEMONE's last invitational WTOR says:

50 DVRM01E DXT DATA EXTRACT MANAGER DEMONE IS READY TO ACCEPT OPERATOR COMMANDS

The MVS operator would enter:

R 50, CHANGE RUNMODE=TERM

Suppose, however, you want an MVS operator to:
- Change DEMONE's run mode from terminating to continuous run mode
- Run DEMONE in continuous run mode for 120 minutes (two hours)

And assume that DEMONE's last invitational WTOR says:

51 DVRM01E DXT DATA EXTRACT MANAGER DEMONE IS READY TO ACCEPT OPERATOR COMMANDS

The MVS operator would enter:

R 51, CHANGE RUNMODE=120

Suppose, finally, that DEMONE has run in continuous run mode for 60 minutes. You now want it to run for only 30 more minutes rather than another 60 minutes to reach the original specified value of 120. Assume that DEMONE's last invitational WTOR says:

52 DVRM01E DXT DATA EXTRACT MANAGER DEMONE IS READY TO ACCEPT OPERATOR COMMANDS

The MVS operator would enter:

R 52, CHANGE RUNMODE=30

Changing a DEM's polling interval
To change a DEM's polling interval, an MVS operator uses the CHANGE command's POLLINTV keyword. Assume you want DEMONE's polling interval changed from its current length to 600 seconds (10 minutes) and that DEMONE's last invitational WTOR says:

53 DVRM01E DXT DATA EXTRACT MANAGER DEMONE IS READY TO ACCEPT OPERATOR COMMANDS
The MVS operator would enter:

R 53, CHANGE POLLINTV=600

**Changing a DEM's priority range**
To change a DEM's priority range, an MVS operator uses the CHANGE command's PRILIM keyword. Assume you want an MVS operator to change DEMONE's priority range from whatever it is currently to 0 through 255 and that DEMONE's last invitational WTOR says:

54 DVRMO1E DXT DATA EXTRACT MANAGER DEMONE IS READY TO ACCEPT OPERATOR COMMANDS

The MVS operator would enter:

R 54, CHANGE PRILIM=(0,255)

**Note:** An MVS operator must always enter both a low value and a high value when changing a DEM's priority range. If, in the above example, the DEM's current priority range is 100 through 255, the MVS operator would still enter:

R 54, CHANGE PRILIM=(0,255)

**Changing a DEM's output limit**
To change a DEM's output limit, an MVS operator uses the CHANGE command's OUTLIM keyword. Assume you want an MVS operator to change DEMONE's output limit from whatever it is currently to 10000 rows of output for each extract request and that DEMONE's last invitational WTOR says:

55 DVRMO1E DXT DATA EXTRACT MANAGER DEMONE IS READY TO ACCEPT OPERATOR COMMANDS

The MVS operator would enter:

R 55, CHANGE OUTLIM=10000

If you make the output limit lower than that specified for extract requests currently eligible for execution, those requests won't be processed. Extract requests eligible for execution get queued again so those having an output limit currently too high do not process until the output limit changes once more.

As another example, assume you want an MVS operator to change DEMONE's output limit so there is no limit of output rows it writes for each extract request and that DEMONE's last invitational WTOR says:

56 DVRMO1E DXT DATA EXTRACT MANAGER DEMONE IS READY TO ACCEPT OPERATOR COMMANDS
The MVS operator would enter:

```
R 56, CHANGE OUTLIM=NONE
```

This allows other requests previously not eligible for execution to now execute.

**Changing a DEM's debug level**

To change a DEM's debug level, an MVS operator uses the CHANGE command's DEBUG keyword. Assume you want an MVS operator to change DEMONE's debug level from what it is currently to a debug level of 4 and that DEMONE's last invitational WTOR says:

```
57 DVRM01E DXT DATA EXTRACT MANAGER DEMONE IS READY TO ACCEPT OPERATOR COMMANDS
```

The MVS operator would enter:

```
R 57, CHANGE DEBUG=4
```

**Changing the routing for operator messages sent to an MVS console**

To change the routing code by which a DEM sends operator messages to an MVS console, an MVS operator uses the CHANGE command's OROUTE keyword. Assume that an MVS operator wants to change the routing code by which DEMONE sends operator messages to an MVS console from the code by which they are currently sent to another routing code and that DEMONE's last invitational WTOR says:

```
58 DVRM01E DXT DATA EXTRACT MANAGER DEMONE READY TO ACCEPT OPERATOR COMMANDS
```

The MVS operator would enter:

```
R 58, CHANGE OROUTE=1
```

**Recovering from a DEM error**

When a DEM starts processing an extract request, it marks the extract request entry in the EXTLIB as *in-use*. When a DEM finishes processing the extract request, it deletes the extract request entry from the EXTLIB unless it is flagged as TYPE=PERSIST on the SUBMIT command. If a DEM abnormally ends or fails while processing an extract request, it, generally, leaves the extract request in-use.

To process an extract request left in-use, you must run a DEM having the same INITDEM NAME as the DEM that abnormally ended or failed. The JES internal reader discards output from the DEM that results from abnormal end or failure to process an extract request. If the DEM has already sent some output to any other output device or to a sequential data set, you should have that output, respectively, discarded.
Whenever a DEM stops executing, it returns a DEM completion code and, as discussed in the remaining sections of this appendix, possibly physical sequential, VSAM, generic data interface (GDI), or IMS/VS DL/I error and reason codes. There are also messages in DXTPRINT detailing the problem(s) met (unless the DEM has problems writing to DXTPRINT). The following table shows the meanings of the DEM completion codes.

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal completion, no errors met.</td>
</tr>
<tr>
<td>4</td>
<td>Noncritical errors met. Other messages detail the problems the DEM met.</td>
</tr>
<tr>
<td>8</td>
<td>Critical errors met. Other messages detail the problems that terminated the DEM.</td>
</tr>
<tr>
<td>12</td>
<td>Unable to open DXTPRINT data set. No DEM message is provided.</td>
</tr>
<tr>
<td>16</td>
<td>Unable to acquire storage for DEM initialization. No DEM message is provided.</td>
</tr>
<tr>
<td>20</td>
<td>Insufficient storage in DEM internal automatic storage pool. A message provides more information. See the <em>DataRefresher Diagnosis Guide</em>.</td>
</tr>
<tr>
<td>24</td>
<td>Error writing to DXTPRINT data set. No DEM message is provided, although a system message may be present.</td>
</tr>
<tr>
<td>28</td>
<td>Invalid completion code passed to ‘emergency exit’ module internally. A message provides more information. See the <em>DataRefresher Diagnosis Guide</em>.</td>
</tr>
<tr>
<td>32</td>
<td>Invalid logical record length for DXTPRINT data set. An LRECL of 121 must be used.</td>
</tr>
<tr>
<td>36</td>
<td>Unknown environment.</td>
</tr>
</tbody>
</table>

If a DEM returns a completion code other than 0 or 4, someone should respond to it. Someone must respond to any physical sequential, VSAM, GDI, or IMS DL/I error and reason codes before the DEM can continue to run or before anyone can restart it.

**Recovering from FDTLIB READ errors**

If an error occurs while a DEM is reading from the FDTLIB, the DEM passes on the appropriate VSAM reason and return codes, and, if the error is serious, immediately terminates. It does not, however, leave any extract requests in-use.

**Recovering from EXTLIB READ errors**

If an error occurs while a DEM is reading from the EXTLIB, the DEM passes on the appropriate VSAM reason and return codes, and, if the error is serious, immediately terminates. It does not, however, leave any extract requests in-use.
Recovering from EXTLIB WRITE errors
If an error occurs while a DEM is writing to the EXTLIB, the DEM passes on the appropriate VSAM reason and return codes, and, if the error is serious, immediately terminates.

WRITE to EXTLIB errors can occur when a DEM tries to delete an extract request from the EXTLIB after processing it or when canceling an extract request or when trying to change the status of an extract request from in-use to available for processing. In any case, error messages provide a system programmer sufficient information necessary to delete the appropriate extract request records from EXTLIB. You may need to contact the DataRefresher diagnostician for help. For example, if you get a message saying that a header record for extract request XYZ remains in EXTLIB, you can use an OS/VS2 Access Method Services program service to remove that header record.

Recovering from database or data file OPEN errors
If an error occurs while a DEM is opening a database or data file, error messages should provide a system programmer sufficient information on how to correct the error. The DEM's current batch of extract requests are requeued in the EXTLIB for complete processing by any eligible DEM.

Note: If a GDI exit is responsible for opening the database or file, the GDI exit supplies to DataRefresher:

- A return code that controls DataRefresher's action
- An appropriate error message (optional)

Recovering from database or data file READ errors
If an error occurs while a DEM is reading a database or data file, error messages should provide a system programmer sufficient information on how to correct the error. The DEM, however, does not requeue its current batch of extract requests under this circumstance; instead, it deletes them from the EXTLIB.

Note: If a GDI exit is responsible for reading the database or file, the GDI exit supplies to DataRefresher:

- A return code that controls DataRefresher's action
- An appropriate error message (optional)

Recovering from DXTOUT data set WRITE errors
If an error occurs while a DEM is writing the output for a given extract request to a DXTOUT data set, the DEM requeues that extract request, decreases by one the maximum number of extract requests in a batch that it can process, and does not attempt to use that DXTOUT data set again during its current execution. If this leaves a DEM with no DXTOUT data sets to write to, the DEM terminates.
Chapter 11. Protecting your General Data Extract data

This chapter provides information on how to protect your data resources from unauthorized use when extracting data with the General Data Extract feature.

- "Protecting individual data descriptions" describes how to protect data descriptions using the Resource Access Control Facility (RACF).
- "Controlling access to the FDTLIB and EXTLIB data sets" on page 129 describes how you can protect your FDTLIB and EXTLIB data sets.
- "Preventing unauthorized use of the UIM and DEM" on page 130 describes how to control access to the UIM and DEM to prevent unauthorized use of the resources that are used by these programs.
- "Protecting exit routines" on page 133 describes how to protect the load libraries at your site that hold exit routines.
- "Protecting source data" on page 133 describes how to protect sensitive source data with absolute protection.
- "Data-protection schemes for single or multiple FDTLIBs" on page 131 describes how you can use RACF to protect your DataRefresher sources for administration schemes that use single or multiple FDTLIBs.
- "Controlling DataRefresher OS/2 access to host resources" on page 134 describes how you can protect data sets and other host resources when using DataRefresher OS/2.

This chapter assumes you have RACF installed and that you are familiar with using it. For information on RACF, see Resource Access Control Facility (RACF) General Information and Resource Access Control Facility (RACF) Command Language Reference.

Protecting individual data descriptions

The UIM has a built-in capability to protect the individual data descriptions in the FDTLIB. Specifically, the UIM can protect system-wide the names used for DXTFILE descriptions, DXTPSB descriptions, and DXTVIEW descriptions, thereby protecting the data descriptions from unauthorized use. You can protect selected data descriptions with this UIM protection scheme.

To protect data, you must define and make active to RACF the resource member class DVRDEFN. To use the resource member class DVRDEFN, you must add the DVRDEFN class name to the following two tables:

- Class Descriptor Table (CDT)
- MVS router table

To protect the DataRefresher data descriptions in the FDTLIB, the names specified on the CREATE DXTFILE and CREATE DXTVIEW commands must conform to the RACF naming conventions established for your site.

DataRefresher supports the system authorization facility (SAF) protocol. For information on creating the DVRDEFN class, see the appropriate DataRefresher program directory. For additional information, see Resource Access Control Facility (RACF) System Programming Library.
The class DVRDEFN is a resource member class, not a resource group class. Each member of the class represents either a DXTFILE description, a DXTPSB description, or a DXTVIEW description. As shown in Table 18 on page 124, the format of the name indicates the type of entry represented. The first part of the name is the resource; the second part is either the DXTFILE, DXTPSB followed by the DXTPCB, or DXTVIEW name.

<table>
<thead>
<tr>
<th>Table 18. DVRDEFN member name formats</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Member name format</strong></td>
</tr>
<tr>
<td>DXTF.dxtfile</td>
</tr>
<tr>
<td>DXTP.dxtpsb.dxtpcb</td>
</tr>
<tr>
<td>DXTV.dxtview</td>
</tr>
</tbody>
</table>

**Note:** Do not include embedded blanks in the member name, unless the name is in double quotes.

The authority scheme for DVRDEFN is essentially one-level. You either have or do not have access to a given DXTFILE description, DXTPSB description, or DataRefresher view description whose name has been added to DVRDEFN. If you do not have access, you can do nothing with the description through DataRefresher commands while the DVRDEFN capability is active.

You can have either explicit or implicit DVRDEFN access to a given description:

- **Access is explicit if the description name was added to DVRDEFN and you were authorized to access the description.** According to the member names given in Table 18, if you were authorized to access the DVRDEFN resource DXTV.PSEQDEPT, you have explicit access to the view named PSEQDEPT. To have explicit DVRDEFN access, a user's RACF access level must be at least READ, although it may be also UPDATE, CONTROL, or ALTER.

- **Access is implicit if the description name is not in DVRDEFN.** According to the member name examples in Table 18, if a DXTVIEW is named NEWVIEW1 rather than PSEQORG, everyone automatically has implicit access to the DXTVIEW named NEWVIEW1. Access is also implicit if RACF is not installed, or if RACF is installed but the DVRDEFN class either was not defined or is not active.

Implicit access is a way to create and use data descriptions that are outside the DVRDEFN authority scheme. A description not written as a DVRDEFN resource is treated as if the DVRDEFN class never existed.

Whether your DVRDEFN access is explicit or implicit, you have the same access privileges to the description. From the viewpoint of the UIM, explicit and implicit access authority are identical.

Using DVRDEFN to protect data can make that data available to a broader audience. If you need absolute protection for your data, see “Protecting source data” on page 133.
**Note:** When users gain explicit DVRDEFN authority to DXTVIEWs, those users, in effect, can access data to which they may not have access through RACF.

For example, suppose User X is given DVRDEFN access to a particular DXTVIEW, VIEWA. VIEWA pertains to FILEA that is owned and RACF-protected by User Y, and it contains fields with highly sensitive data. User Y is authorized to see the source data underlying FILEA, and is authorized to run the DEM. User X is neither authorized to see the source data underlying FILEA, nor authorized to run the DEM. Thus, after User X submits an extract request naming VIEWA, and when User Y runs the DEM to process requests using FILEA as source data, User X can extract data which User X normally would not be able to access.

### Using DVRDEFN to protect data

This section describes what DVRDEFN authority is required to perform a specific DataRefresher operation (for example, to print a particular DXTFILE description). You can perform the DataRefresher operation if you have DVRDEFN authority and you meet all the non-DVRDEFN requirements, such as appropriate RACF authority to the FDTLIB and access to the UIM (see “Controlling access to the FDTLIB and EXTLIB data sets” on page 129 and “Preventing unauthorized use of the UIM and DEM” on page 130).

### Creating data descriptions under control of DVRDEFN

The following is required for creating data descriptions under control of DVRDEFN. (You should add a member to DVRDEFN before the description that it represents is created, thereby controlling who can create the description.)

- Creating a DXTFILE description requires DVRDEFN access to that DXTFILE description
- Creating a DXTPSB description requires DVRDEFN access to each of the DXTPCB descriptions in that DXTPSB description
- Creating a DXTVIEW description requires DVRDEFN access to the view, along with DVRDEFN access to the underlying DXTFILE or DXTPCB description

### Using and maintaining data descriptions under control of DVRDEFN

The following is required for using and maintaining data descriptions under control of DVRDEFN:

- Using a DXTVIEW description in an extract request requires DVRDEFN access to the view (but not for the underlying DXTFILE or DXTPCB description).
- Printing or punching a DXTFILE or DXTVIEW description requires DVRDEFN access to the description.
- Printing or punching an entire DXTPSB description requires DVRDEFN access to each of the PCBs in the DXTPSB description (If you try to print a PSB that contains some PCBs you are not authorized to see, only those PCBs to which you have DVRDEFN access are printed).
- Printing or punching a DXTPCB description requires DVRDEFN access to that DXTPCB description.
• Deleting any DXTFILE or DXTPSB description requires the DVRDEFN authority to re-create the description. Thus:
  – Deleting a DXTFILE description requires DVRDEFN access to the description.
  – Deleting a DXTPSB description requires DVRDEFN access to each of the DXTPCB descriptions in the DXTPSB description.

• Deleting a DXTVIEW description does not require the DVRDEFN authority to re-create the description. Thus:
  – Deleting a view of a file requires DVRDEFN access to the view, but not to its underlying DXTFILE description.
  – Deleting a view of an IMS DL/I database requires DVRDEFN access to the view, but not to its underlying DXTPCB description.

No DataRefresher program can change the DVRDEFN members, and all changes to these members must be done outside DataRefresher. RACF and DVRDEFN authority can be granted and revoked with RACF commands only.

Maintaining the DVRDEFN protection scheme
One or more users at your site need to administer the DVRDEFN capability. These DVRDEFN administrators probably create and maintain the DataRefresher file, DXTPSB, and DXTVIEW descriptions as well.

To identify DVRDEFN administrators to RACF, change the CLAUTH parameter in the DVRDEFN administrator's profile. The RACF administrator or someone with the RACF SPECIAL authority attribute designates the DVRDEFN administrators as follows:

• Changes the user profile of each DVRDEFN administrator by adding DVRDEFN after the CLAUTH parameter.

• Changes the user profile of the first DVRDEFN administrator. Then, the new DVRDEFN administrator can identify other DVRDEFN administrators by changing the CLAUTH parameter in their user profiles. The first DVRDEFN administrator, however, must own the profiles of any user whose CLAUTH parameter the administrator wishes to change (The owner of a profile, normally the RACF administrator, is indicated in the OWNER parameter of the user's profile).

DVRDEFN administrators can define resources to the DVRDEFN class with the RACF RDEFINE command, and they can use the PERMIT command to let others (either other DVRDEFN administrators or other users) use resources that they have defined. In this way, DVRDEFN administrators control their own groups of DVRDEFN resources and who can access those resources.

All DVRDEFN administrators can use the RLIST command to examine a list of all resources defined to DVRDEFN—regardless of who defined the resources. Remember that RLIST yields the names of the data descriptions that are DVRDEFN protected—not the data descriptions themselves. Any administrator can see all the resources protected by DVRDEFN by using the RLIST command. However, an administrator can use the RACF maintenance commands (RALTER, RDELETE, and PERMIT) only on the resources that the administrator defined, unless the owner of other resources granted the ALTER privilege to other administrators.
Example using DVRDEFN to protect data

Suppose you want to create the DXTPSB description named SAMPPSB1 which is part of DataRefresher's IMS sample data. SAMPPSB1 contains one DXTPCB description named SAMPPCB1. You also want to create the views, IMSSTAFF and IMSPROJECT, which are based on SAMPPCB1, and you want to let two nonadministrative users, Arnett and Hybki, use these views.

Although for this example it is assumed that you are working interactively, you can submit a job in batch mode, as well.

1. Log on to TSO.

2. Issue the following RACF RLIST command:

   RLIST DVRDEFN * ALL

   This command produces a list of all the DVRDEFN resources. On this list, look for resources named DXTP.SAMPPSB1.SAMPPCB1, DXTV.IMSSTAFF and DXTV.IMSPROJECT. If such resources exist, you cannot build your DXTPSB description or DXTVIEWs unless you choose different names for your resources or if you, or someone else, first delete the existing like-named entries.

   For this example, however, assume that you find no such names in your list.

3. Issue the following RACF RDEFINE and PERMIT commands:

   ```
   RDEFINE DVRDEFN DXTP.SAMPPSB1.SAMPPCB1 UACC(NONE) 1
   RDEFINE DVRDEFN DXTV.IMSSTAFF UACC(NONE)
   RDEFINE DVRDEFN DXTV.IMSPROJECT UACC(NONE)
   PERMIT DXTV.IMSSTAFF CLASS(DVRDEFN) ID(ARNETT, HYBKI) ACCESS(READ) 2
   PERMIT DXTV.IMSPROJECT CLASS(DVRDEFN) ID(ARNETT, HYBKI) ACCESS(READ)
   ```

   1 The RDEFINE commands define the DVRDEFN resources that represent the DXTPCB description, SAMPPCB1, and the DXTVIEWs, IMSSTAFF and IMSPROJECT. The NONE parameter on the UACC statements indicates that you have the only DVRDEFN access to those data descriptions.

   2 The PERMIT commands give Arnett and Hybki READ access to the DXTVIEWs IMSSTAFF and IMSPROJECT. READ access enables Arnett and Hybki to submit extract requests using these DXTVIEWs and to print these DXTVIEWs after these views are created.

   However, Arnett and Hybki cannot delete these DXTVIEWs, because they were not given ALTER or UPDATE access.

4. Submit a UIM job to create the DXTPSB and DXTVIEW descriptions. You get the greatest degree of protection if you DVRDEFN-protect your data description names before you actually define those data descriptions. Also, because you defined the data descriptions to DVRDEFN, you automatically get DVRDEFN access to those resources.

The RACF administrator must keep the DXTFILE, DXTPSB, and DXTVIEW authorizations of all users current.
Protecting data descriptions created by the DAP

If you are using the Dictionary Access Program (DAP) to generate commands to create data descriptions, you should be aware of the security scheme that protects IBM OS/VS DB/DC Data Dictionary data and DAP output before the output is sent to the UIM.

Sensitive data stored in the data dictionary is protected by the data dictionary's security facility. Each user of the dictionary has an access profile that specifies which data categories and status codes the user can access, and whether that access is for update or read-only purposes.

When the DAP requests data from the data dictionary, the dictionary checks the access profile of the user ID to see if the user ID is authorized to access the requested dictionary data. If the user is not authorized, the dictionary does not return the requested data, and the DAP, after issuing a message, attempts to continue processing. However, the DataRefresher commands that the DAP generates will not have the data that the user was not authorized to access. For more information about the data dictionary security facility, see IBM OS/VS DB/DC Data Dictionary Administration and Customization Guide.

If your site plans to use the DAP to generate sensitive DataRefresher data descriptions, you need to ensure the security of your DAP jobs. That is, you must ensure that the:

- Dictionary data is properly protected by the dictionary security facility
- DAP output is protected by RACF security

When you run a DAP job, ensure that the requesting user ID is authorized to read all the data required to generate the data description commands. For example, the user ID might need to read PSBs, PCBs related to the PSB, databases related to the PCBs, real segments related to view segments, and elements related to segments and other elements. In addition, the user ID might need to read files (in the DATABASE category), segments related to files, and elements related to segments and other elements.

After you access the dictionary data, you need to protect the DAP output, which is a punched-format data set.

If the output is immediately input to the UIM, the DAP job should be in two steps. In the first step, the DAP generates the DataRefresher commands and places them in a temporary data set. The temporary data set is then passed to the second step, which invokes the UIM to put the descriptions in the FDTLIB.

In addition to having access to the dictionary data, the user ID running the DAP job must also be RACF-authorized to run the UIM and to add these descriptions to the FDTLIB (if both the UIM and FDTLIB are RACF-protected).

If the DAP output is not immediately entered into the FDTLIB, for example, if it must be manually edited before being added to the DataRefresher library, it should be placed in a RACF-protected data set until the output can be changed and submitted to the UIM. Again, however, the user ID running the DAP job must be authorized not only to access dictionary data, but also RACF-authorized to access the data set containing the DAP output.
Controlling access to the FD铊LIB and EXTLIB data sets

You might want to control access to your FD铊LIBs and EXTLIBs for the following reasons:

- To protect the data sets from someone using a non-DataRefresher program either to read secured information from one of the libraries, or to put an unauthorized definition or request into one of the libraries.
- To protect the integrity of the data sets which might be jeopardized by unrestricted access to them.

The RACF administrator must control each user's access to your FD铊LIBs and EXTLIBs. (Like DVRDEFN administration, RACF administration requires that the user be authorized to maintain RACF. This user is referred to as the RACF administrator.)

As a RACF administrator, you should provide users with access only to the data sets they need. You can control access to these data sets through RACF grants of the READ and the UPDATE privileges.

Protecting FD铊LIBs

To control access to FD铊LIBs, you should grant READ authority to users who do either of the following:

- Submit extract request jobs
- Submit requests to cancel extract requests or check on the status of extract requests

You should grant UPDATE authority to any user who creates and maintains data descriptions in an FD铊LIB.

Users who change DXTRFILE or DXTPSB descriptions also require UPDATE authority if they need to update any DXTVIEW descriptions that correspond to the modified DXTRFILE or DXTPSB descriptions. If these users cannot update the DXTVIEW descriptions, subsequent users of the views will have problems when they run their extract requests.

Note: Users who have UPDATE authority to an FD铊LIB can create new data descriptions and DXTVIEW descriptions over the data, even over data that they may not have RACF authority to see. In this case, you need to consider providing absolute protection of source data with RACF. See "Protecting source data" on page 133 for details.

Protecting EXTLIBs

To control access to EXTLIBs, you should grant UPDATE authority to those users who:

- Are authorized to submit extract requests
- Run the DEM to process extract requests

DataRefresher users who can run the DEM need UPDATE authority because the DEM modifies an EXTLIB in the course of a run. For example, the DEM replaces a successfully run request with a marker to be used in satisfying STATUS commands, and it removes these markers from an EXTLIB after a predefined length of time.
If you are concerned that an unauthorized user may study some extract requests to submit an unofficial request, you may want to limit granting UPDATE authority for a particular EXTLIB. In this way, only those users with UPDATE authority could access that particular EXTLIB.

Because all DataRefresher users need access to some EXTLIB, you may choose not to limit access to an EXTLIB. You would rely on DVRDEFN protection of the individual data descriptions and RACF protection of the data sets to keep the DataRefresher resources secure. However, before you decide not to protect the EXTLIBs at your site, be sure that your resources are adequately protected by other security mechanisms.

The JCL procedure supplied with DataRefresher, DXTINRUN, which runs the UIM and DEM together, can offer some protection for your extract requests. The procedure automatically creates an EXTLIB, which it uses only during that run of the programs; this EXTLIB holds only the extract requests submitted during that run of the UIM. The next time DXTINRUN is called, it erases the first EXTLIB and creates another (with the same name) to hold the new batch of extract requests. While DXTINRUN creates a temporary EXTLIB for the extract request, it uses data descriptions stored in the permanent FDTLIB(s) to process those requests. Therefore, you may still want to consider some type of security scheme for protecting FDTLIBs. (For more information about the DXTINRUN JCL procedure, see DataRefresher MVS and VM User's Guide.)

Preventing unauthorized use of the UIM and DEM

You may want to control access to the UIM and DEM to prevent unauthorized use of resources that are used by these programs. You can restrict access to the DataRefresher programs by placing each component into a private program library. (To do this, copy the load modules of the programs. The name of the components of the UIM are:

- DVRTIO000
- DVRRL0000
- DVRU0000

The name of the components of the DEM are:

- DVRTB0000
- DVRIO0000
- DVRRL0000
- DVRX0000

After you place the UIM and DEM in private program libraries, grant RACF READ access for the libraries to those who use the programs.

To further protect the DEM, you may choose to limit users who can run the DEM (their user IDs are specified in the DEM job).

If you have sensitive data and want to be the only user with access to that data, in addition to RACF-protecting the data, you can create a private program library for a DEM and give only yourself access to that DEM.
Data-protection schemes for single or multiple FDTLIBs

You can use RACF to protect your DataRefresher sources whether you choose to implement an administration scheme that uses single or multiple FDTLIBs. Generally, the scheme for using a single FDTLIB places the responsibilities of creating and protecting data descriptions on an administrator and is designed for maximum control. On the other hand, the scheme for using multiple FDTLIBs distributes the responsibilities of creating and protecting data descriptions to various user groups. You might find that either of these two protection schemes, or that some combination of these protection schemes works best. There are many possibilities of protection schemes. The following discussions are just two examples.

Using a single FDTLIB

Assume that your site provides a single FDTLIB data set and a single EXTLIB data set for everyone to use. Also, assume you want to use DVRDEFN to protect data descriptions and RACF to protect the UIM and DEM.

Under this data-protection scheme, your users' responsibilities are grouped as follows:

- Administrators protect and create the data descriptions in the FDTLIB, and control access to the DXTVIEWs that they create. To perform these tasks, administrators need:
  - Authority to create and maintain the DVRDEFN resources through RACF commands
  - READ access to the UIM program library
  - UPDATE access to the FDTLIB
  - UPDATE access to the EXTLIB (if they submit extract requests)

Administrators explicitly protect every DXTFILE or DXTPSB description that they create. That is, they RACF-define the corresponding DVRDEFN resources, and limit access to these descriptions to themselves and other administrators. Administrators explicitly protect every DXTVIEW they create that cannot be shared with all users, and they grant DVRDEFN access to authorized users. Administrators may or may not explicitly protect DXTVIEWs that can be used by everyone.

- Users can formulate extract requests and instruct DataRefresher to run them, but they cannot create or delete data descriptions in the FDTLIB. To perform these tasks, users need:
  - READ access to the FDTLIB and the UIM program library
  - UPDATE access to the EXTLIB

Users are not given access to the DEM program library. They have DVRDEFN access only to the DXTVIEW descriptions to which the administrator grants them access.

- A user who can run the DEM is often a DataRefresher administrator. The user ID of the DataRefresher user who has authority to run the DEM is specified on the DEM job when the job is run. To run the DEM, users need:
  - READ access to the FDTLIB and the DEM program library.
  - UPDATE access to EXTLIB.
- Authority to read data from the source non-IMS files and IMS DL/I databases. (This authority is different from DVRDEFN access to the DXTFILE, and DXTPSB descriptions.)

This data-protection scheme is designed for maximum control over the user. Under this scheme a user must depend on an administrator for data descriptions and the DVRDEFN authority to use specific DXTVIEWs. A user can print or punch these authorized DXTVIEWs, but not the underlying data descriptions. Users cannot modify these data descriptions because they only have RACF READ privileges on the FDTLIB and do not have DVRDEFN access to the data descriptions. If a user cannot run an extract request because the underlying data description has been changed, that user depends on the administrator to regenerate the corresponding DXTVIEW descriptions.

**Using multiple FDTLIBs**

Assume that your site distributes the tasks of creating and protecting data descriptions to specific user groups. Also, assume that your site provides these user groups with multiple FDTLIBs and EXTLIBs.

With this data-protection scheme, users are divided into groups based on common interest in, and access to, the available data. To perform its DataRefresher operations, each user group owns a single FDTLIB and EXTLIB. In each group, users with UPDATE access to the FDTLIB create the data definitions for that group. Those users that have READ or UPDATE access to the FDTLIB and UPDATE access to the EXTLIB submit the extract requests. One or more users in each group has access to the DEM, and these users control the DEM. Users have access only to their group's own FDTLIB and EXTLIB.

If DVRDEFN protection is not used, each user group is independent of other user groups and, therefore, responsible for its own operation. Individual groups can negotiate with the database administrators for access to IMS DL/I, VSAM, and other data structures, and each group can formulate, for example, its own security rules and naming conventions.

However, if DVRDEFN protection is used, then the group independence previously described is destroyed because the DVRDEFN class resources apply equally to every FDTLIB. If, for example, there are two or more DXTVIEW descriptions with the same name in various FDTLIBs, you either have DVRDEFN access to all of them or to none of them, and the same is true for all DXTFILE and DXTPSB descriptions. That is, the DVRDEFN protection scheme protects the names of the data descriptions and not their locations. To avoid this situation, you could use a system-wide naming convention. Each group assigns a unique prefix for the names of the data descriptions it creates. That way, a group may create and DVRDEFN-protect its data descriptions without inhibiting another group in any way.
Protecting source data

If you or others at your site have highly sensitive data, you might want to provide the data with absolute protection. As the owner of highly sensitive data, you can create and submit the data descriptions that support it and store them in a private FDTLIB to which only you have access.

In addition to owning your own FDTLIB, you should own your own EXTLIB, for which only you have UPDATE access, because your EXTLIB contains extract requests that reference the sensitive data. You should also RACF-protect your sensitive source data, EXTLIB, and FDTLIB, and give access only to yourself. You will also want RACF authorization to run the DEM, to process any job that you submit, thereby ensuring that only authorized users (in this case, you) see the sensitive data.

By following these steps, you will not need to DVRDEFN-protect your data descriptions. DVRDEFN cannot offer any more protection than you already have by owning (and RACF-protecting) your own FDTLIB and EXTLIB.

Protecting exit routines

Because you can protect the UIM and DEM from unauthorized use, you might also want to protect the load library (or libraries) at your site that holds the exit routines. By granting RACF READ access to the library containing the exit routines, you can let certain users view the exit routines. RACF UPDATE access lets users update or change the exit routines. Users without either RACF authority cannot access the load libraries containing the exit routines.

To protect the exit routine load libraries, the RACF administrator must control access to the load libraries.
Controlling DataRefresher OS/2 access to host resources

DataRefresher OS/2 allows two levels of security for your DataRefresher resources:

- Host security
- Advanced Program-to-Program Communication (APPC) security

Host security

The data protection schemes for operating DataRefresher on the host system also protect your data when you operate DataRefresher OS/2 from a workstation. On the host system, you can control access to source data, data descriptions, the UIM, the DEM, and FDTLIB and EXTLIB data sets using RACF or an equivalent security product.

To enable a user to access host resources from DataRefresher OS/2, the user's DataRefresher user ID and password must be registered with Communications Manager/2. For information on registering user IDs and passwords with Communications Manager/2, see *IBM Communications Manager/2 Version 1.0: Configuration Guide*.

APPC security

In addition to the security provided on the host system, DataRefresher OS/2 provides for APPC security. The APPC security mechanism controls workstation access to the host machine and to the host DataRefresher transaction program (TP).

APPC provides for the following security measures:

- **APPC LU-LU partner verification** checks that the logical unit (LU) defined for a workstation is authorized to communicate with the host system LU
- **APPC end-user verification** checks that the application program being run on a workstation is authorized to call DataRefresher host functions

The APPC security mechanism depends on RACF or an equivalent product being installed at your site.

For information on setting up APPC security, see *MVS/ESA Planning: APPC Management*.
Chapter 12. Administering the Dictionary Access Program

If you choose to use the DataRefresher Dictionary Access Program (DAP) at your site, as an administrator you may need to:

- Supply DAP JCL for DataRefresher users who use the data dictionary
- Run the DAP
- Establish guidelines for data dictionary maintenance

This chapter provides information you need to administer the DAP.

- "What the Dictionary Access Program does" describes how the DAP works and how you provide input to the DAP.
- "What information the DAP needs from the data dictionary" on page 137 will help you establish guidelines for maintenance of the IBM OS/VS DB/DC Data Dictionary to ensure that the DAP finds the information it requires.

What the Dictionary Access Program does

If you have the IBM OS/VS DB/DC Data Dictionary installed, you can automate the job of creating DXTFILE and DXTPSB descriptions using the DAP. Figure 39 illustrates how the DAP works with the data dictionary.

![Diagram showing DAP operation](image)

1 The DAP uses the Program Access Facility of the data dictionary to create data descriptions. (Thus, you should have experience of using the data dictionary before you use the DAP.)

2 When you identify files and PSBs to the DAP, the DAP generates DataRefresher UIM commands to create data descriptions for the designated files and PSBs. The DAP then writes a file composed of the generated UIM commands. When used as the DXTIN data set on a run of the UIM, this file creates the requested data descriptions.

Before you use this file, however, you may want to edit it. For example, you can delete information that you do not want to appear in the data descriptions, such as FIELD statements (on the DataRefresher CREATE commands) for data that you do not want users to access. You may also need to provide information that was not available to the DAP when it was run.

3 The data descriptions created with the DAP are submitted to the UIM, which saves them in the FDTLIB.
Writing input for the DAP

To provide input to the DAP, you use the EXECUTE command through either of the following:

- A batch job
- The Administrative Dialogs

The DAP passes the EXECUTE command to the data dictionary to request data definition information.

To provide input to the DAP through a batch job, you put the EXECUTE command in the DDINPUT data set. You do not need to limit your input to the DAP to a single EXECUTE command. You can have more than one EXECUTE command as well as other data dictionary commands.

Using the Administrative Dialogs, you can build the EXECUTE command. The Administrative Dialogs combine the command with JCL that will run the DAP. You can edit this job before submitting it, to add more EXECUTE commands or other data dictionary commands.

For complete EXECUTE command syntax information, see DataRefresher Command Reference. For information on using the EXECUTE command in a batch job and in the Administrative dialogs, see the DataRefresher MVS and VM User's Guide.

Writing JCL to run the DAP

You can write JCL that calls the data dictionary through IMS. You can combine this JCL for running the DAP with the JCL for running the UIM to make one job containing two steps:

- Step 1: Generate data descriptions
- Step 2: Submit the data descriptions to the UIM

For details on writing JCL to run the DAP, see DataRefresher MVS and VM User's Guide.

Reviewing the DAP output

Review your DAP output for unwanted, missing, and possibly erroneous information, before you use it to create data descriptions.

For example, output for a DXTFILE or DXTPSB description might describe more segments, fields, or PCBs than you want to describe. You can eliminate this information with an editor. Information can be missing from DAP output because the dictionary allows its users to enter information about a subject incrementally. Thus, subject and relationship data that the DAP requires could be incomplete. Also, information can be erroneous if the dictionary is not maintained properly or information is not entered correctly.

To avoid unwanted, missing, and erroneous information in your DAP output, establish guidelines for data dictionary maintenance to ensure that the DAP finds the information it requires.
What information the DAP needs from the data dictionary

To ensure that the DAP finds all the information it needs in the dictionary databases, you must know what information the DAP expects and where it expects to find it.

This section indicates the type of information the DAP needs from the data dictionary. To benefit from this section, you should have a basic knowledge of the data dictionary.

How the DAP determines the values of NAME keywords

Every command generated by the DAP (except CREATE) has a NAME keyword to identify the entity that the command represents. The DAP determines the identifying name as follows:

**DXTFILE description names**: The DAP uses the user's subject name, as specified in the EXECUTE command.

**DXTPSB description names**: The DAP uses the user's subject name, as specified in the EXECUTE command.

**DXTPCB description names**: If the primary name for the PCB or one of its aliases matches the PSB name, the DAP forms the DXTPCB description name by appending the 3-digit occurrence number of the PCB to the first five characters of the PSB name. If the PSB name is fewer than five characters long, the occurrence number is appended to the entire name.

If neither the primary name for the PCB nor any of its aliases matches the PSB name, the DAP uses the primary name for the PCB.

**IMS segment names**: For IMS DL/I segments, the DAP:

- Uses the segment's primary user name if the subject code is A.
- Uses an alias with the subject code A, if the subject code is not A.
- Writes a message to DDLIST (data dictionary data set that contains EXECUTE commands and run messages) and flags the NAME keyword of the SEGMENT statement with asterisks, if an alias you specified does not exist.

The DAP generates an internal segment for each field in the structure being processed if that field has associated COBOL OCCURS information or PL/I information stored in the data dictionary. If both types of information are present, the DAP gives preference to the COBOL OCCURS clause.

However, some circumstances may prevent the DAP from generating such segments. For example:

- If COBOL OCCURS information is incrementally defined, it may be stored without first being validated. Consequently, the DAP might not be able to accurately interpret the meaning of this information.
- A structure may contain nested fields, each containing COBOL OCCURS information, which normally would indicate the generation of internal segments. This does not present a problem for the DAP if all the internal segments are of fixed length or if the lowest level of nested internal segments is of variable length. However, the DAP does not have the necessary information to produce an accurate structure if a variable-length
internal segment is the parent of either a fixed-length internal segment or another variable-length internal segment.

- If you specify not to use subfields (SUBFIELDS = N) on the EXECUTE command, the DAP will process only those fields directly related to the segment. Consequently, any subfield that contains COBOL OCCURS or PL/I information, and is not directly related to the segment, will not be processed by the DAP. Thus, the UIM statements that are generated may not accurately describe the format of the data stored in the file or database.

- In certain instances, the DAP cannot determine if a field is in a nested structure of internal segments. This situation commonly occurs when a subfield of a structure is related to, for example, its grandparent field as well as its parent. The DAP always produces its output statements from the highest relationship level and may be unaware if a field is part of a nested structure.

- To ensure that the parent of an internal segment is properly generated by the DAP, the physical relationship between the field that contains the COBOL OCCURS information and a segment needs to be established.

**Segment names in a DXTFILE description:** For segments in a DXTFILE description, the DAP uses the segment's primary user name if the name satisfies the naming conventions for segments. If the user name does not conform to these conventions, the DAP selects an alias that does conform. If no such alias exists, the DAP writes a message to this effect to DDLIST and flags the NAME keyword of the SEGMENT statement with asterisks.

For information on DataRefresher naming conventions, see *DataRefresher Command Reference*.

**Field names:** The DAP uses the same procedure for selecting field names that it uses for selecting segment names.

### How the DAP determines the values of other keywords

The DAP picks the value of every keyword, except the NAME keyword, from a dictionary value determined by the type of keyword. The relationship between keyword and value is shown in Table 19. Note that the DD1 value has an ID keyword of DSGNMBR=1.

<table>
<thead>
<tr>
<th>DataRefresher statement</th>
<th>Keyword</th>
<th>Dictionary category/relationship</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXTPCB</td>
<td>DBACCESS</td>
<td>DATABASE</td>
<td>DBACCS</td>
</tr>
<tr>
<td>DXTFILE</td>
<td>ACCESS</td>
<td>DATABASE</td>
<td>INDACCS</td>
</tr>
<tr>
<td>DXTFILE</td>
<td>DDNAME</td>
<td>DATABASE</td>
<td>DD1</td>
</tr>
<tr>
<td>SEGMENT</td>
<td>BYTES</td>
<td>SEGMENT</td>
<td>BYTES</td>
</tr>
<tr>
<td>SEGMENT</td>
<td>FORMAT</td>
<td>SEGMENT</td>
<td>MINBYTES, MAXBYTES</td>
</tr>
<tr>
<td>SEGMENT</td>
<td>PARENT</td>
<td>PCB WITH SEGMENT</td>
<td>PARENT</td>
</tr>
<tr>
<td>FIELD</td>
<td>TYPE</td>
<td>ELEMENT</td>
<td>TYPE</td>
</tr>
<tr>
<td>FIELD</td>
<td>BYTES</td>
<td>ELEMENT</td>
<td>BYTES</td>
</tr>
<tr>
<td>FIELD</td>
<td>SCALE</td>
<td>ELEMENT</td>
<td>DECIMALS</td>
</tr>
<tr>
<td>FIELD</td>
<td>START</td>
<td>SEGMENT WITH ELEMENT</td>
<td>START</td>
</tr>
</tbody>
</table>

Table 19 (Page 1 of 2). Keyword values
Table 19 (Page 2 of 2). Keyword values

<table>
<thead>
<tr>
<th>DataRefresher statement</th>
<th>Keyword</th>
<th>Dictionary category/relationship</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIELD</td>
<td>SEQFLD</td>
<td>SEGMENT WITH ELEMENT</td>
<td>GENFLD</td>
</tr>
<tr>
<td>FIELD</td>
<td>SEQUENCE</td>
<td>SEGMENT WITH ELEMENT</td>
<td>GENFLD</td>
</tr>
<tr>
<td>FIELD</td>
<td>UNIQUE</td>
<td>SEGMENT WITH ELEMENT</td>
<td>GENFLD</td>
</tr>
</tbody>
</table>

What the DAP retrieves for the DESC keyword
When the DESC keyword is generated on a CREATE DXTPSB or CREATE DXTFILE command, or a DXTPCB, SEGMENT, or FIELD statement, the DAP retrieves only the first 50 bytes of the first description segment encountered for the entity. Thus, if data dictionary segment number 299 is the first segment encountered for entity X, then the first 50 bytes of that description are used.

What the DAP retrieves for the FREQ keyword
When the FREQ keyword is generated on a CREATE DXTPSB or CREATE DXTFILE command, the DAP will retrieve the value from the data dictionary's FREQ keyword field, including two decimal positions.

How the DAP interprets the GENFLD value
The GENFLD value applies to fields in either IMS DL/I databases or non-IMS files. It does not apply to IMS DL/I index fields. The GENFLD value determines whether the SEQFLD, SEQUENCE, and UNIQUE keywords should be generated for a given FIELD statement, and what their values should be if they are generated. The following table shows how GENFLD governs these keywords.

Note: A dash (—) in a column indicates that the corresponding keyword is not written. If a value other than those shown is found in GENFLD, all three keywords are generated with asterisks for their values.

Table 20. How GENFLD governs SEQFLD, SEQUENCE and UNIQUE keywords

<table>
<thead>
<tr>
<th>Field type</th>
<th>GENFLD</th>
<th>SEQFLD</th>
<th>SEQUENCE</th>
<th>UNIQUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>undefined</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Database</td>
<td>blank</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Database</td>
<td>G</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Database</td>
<td>U</td>
<td>R</td>
<td>ASC</td>
<td>Y</td>
</tr>
<tr>
<td>Database</td>
<td>M</td>
<td>R</td>
<td>ASC</td>
<td>N</td>
</tr>
<tr>
<td>Field</td>
<td>undefined</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Field</td>
<td>blank</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Field</td>
<td>G</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Field</td>
<td>U</td>
<td>R</td>
<td>ASC</td>
<td>Y</td>
</tr>
<tr>
<td>Field</td>
<td>M</td>
<td>R</td>
<td>ASC</td>
<td>N</td>
</tr>
<tr>
<td>Field</td>
<td>A</td>
<td>R</td>
<td>ASC</td>
<td>N</td>
</tr>
<tr>
<td>Field</td>
<td>D</td>
<td>R</td>
<td>DES</td>
<td>N</td>
</tr>
</tbody>
</table>

Chapter 12. Administering the Dictionary Access Program 139
How the DAP interprets an alternate PCB

The DAP will create a DBACCESS of HSAM, a dummy segment, and a dummy field for an alternate transaction program (TP) PCB so that the PCB can be used as input to the UIM. Do not use a TP PCB in a DXTVIEW or to extract data.

Possible conflicts in dictionary data usage

The DAP uses dictionary data in the standard way to generate statements for a DXTPSB description. However, the DAP must use some dictionary data—INDACCS and DD1—in a nonstandard manner when it generates the statements for a DXTFILE description. This can produce incorrect results.

In standard dictionary practice, both INDACCS and DD1 apply exclusively to IMS DL/I subjects. If your site also uses them for files, however, that usage could conflict with the normal usage of INDACCS and DD1, and the DAP output might be missing necessary information.
Chapter 13. Managing extract requests

To keep information at your site up to date, you may need to run the same extracts at regular intervals. For example, you may need to extract certain data from an IMS database every week, and so you run the appropriate extract request weekly. While developing a method to run certain extract requests repeatedly, you must always consider your operations environment.

This chapter provides some guidelines you can use to help set up your operations environment for running the same extract requests on a regular basis.

- "Choosing a method for running an extract on a regular basis" provides information on submitting extract requests at preset times and submitting persistent extract requests.

- "Managing data sets containing JCL, JCS, and DataRefresher commands" on page 142 provides information on setting up partitioned data sets (PSDs) and grouping data sets according to use.

- "Updating the EXTLIB with extract requests" on page 143 provides information on using the Access Methods Services (AMS) REPRO command to run your regular extract requests.

Choosing a method for running an extract on a regular basis

If users at your site will run the same extract request at regular intervals, you can set up a method for any of the following situations:

- Your operations department submits the same complete extract job at preset times
- Your operations department submits on persistent extract
- Your operations department submits DEM jobs at preset times
- You, as the DataRefresher administrator, run all the jobs

If you choose to let your operations department submit the same extract job at preset times, the job must do one of the following:

- Submit a persistent extract request. You can specify whether you would like the extract request to remain (persist) in the EXTLIB after processing, by specifying TYPE=PERSIST in the SUBMIT/EXTRACT command. This way you can rerun the extract request without having to recreate it and send it to the EXTLIB.

- Store the extract request at a preset time and flag the extract request as a persistent extract. The job must then submit a job to run the DEM at the preset time.

In either case, the job must be self-sufficient. This means that for a DB2 load job, the output must always be placed in the same table. For output sent to a data set, the output must always be placed in the same data set.
If you choose to let your operations department submit DEM jobs at preset times, as the administrator, you:

- Store the extract requests in the EXTLIB.
- Change the target table name each time, if necessary. Doing so is useful in cases where you want to place similar data in dated tables for comparison purposes. You can change the JCS, and you might decide to include a DB2 table reorganization or sort the input data.

If you choose to run all the extract request jobs, you will avoid having to set up procedures for your operations department to follow. However, if you are away from your job for a period of time, your colleagues may not know which jobs need to be run and when.

**Managing data sets containing JCL, JCS, and DataRefresher commands**

When you submit your data descriptions to the UIM, the UIM stores the data descriptions in the FDTLIB, and they remain there. Usually, you only use the JCL to submit your data descriptions when you initially submit them to the UIM, or when you want to update them.

By default, after an extract request is run by the DEM, that extract request is deleted from the EXTLIB data set. So, you would have to resubmit those extract requests that you want to run again, thereby using the same JCL and JCS files and extract request again and again. This is not necessary if you specify your extract to be persistent. Persistent extract requests remain in the queue until they are canceled.

To help you organize your necessary JCL, JCS, data descriptions, and extract requests, you can either:

- Set up three partitioned data sets (PDSs)
- Group the JCL, JCS, data descriptions, and extract requests in terms of extract frequency

**Creating partitioned data sets**

If you choose to set up three PDSs, you can set them up as follows:

- One PDS in which each member is a JCL file to invoke the UIM or DEM. For example, you can separate the members into three categories in which you write JCL that:
  - Updates the FDTLIB
  - Updates the EXTLIB
  - Invokes the DEM

To help you manage these data sets, you can use a naming convention that reflects this organization of members by beginning the name of each member that:

- Updates the FDTLIB with FDT or F
- Updates the EXTLIB with EXT or E
- Invokes the DEM with DEM or D
You can choose to set up one job with multiple steps, so that the first step places the extract request in the EXTLIB, and the second step runs the DEM.

You should also consider the number of extracts that you want to process in a single DEM run. You can take advantage of the DEM's extract request batching capabilities and put together all requests that refer to the same files in a DEM run. For information on batching extract requests, see "Batching extract requests" on page 108.

- One PDS in which each member is a JCS file.
- One PDS in which each member contains DataRefresher commands. You can set up the PDS for DataRefresher commands similar to the organization discussed for JCL files.

**Note:** Set up procedures to back up the data sets you created to hold JCL, JCS, and DataRefresher commands, as well as the DataRefresher system libraries, the FDTLIB, and the EXTLIB.

**Grouping data sets according to frequency of use**

You can group JCL and JCS files, data descriptions, and extract requests in terms of how frequently you need to run them. For example, you could define three types of UIM jobs:

- Daily
- Weekly
- Monthly

In each of these UIM jobs, you can put the appropriate extract requests and the necessary JCS. You run the DEM in a trailing job step.

Each time you add an extract request, be sure you have:

- The correct USE command
- Sufficient DXTOUTn data definition JCL statements for maximum batching

If, for example, a weekly job gets too large, you can split it into WEEKLY1 and WEEKLY2, where WEEKLY2 contains the job to run the DEM.

**Updating the EXTLIB with extract requests**

If the extract requests you run at regular intervals are fairly constant, avoid resubmitting the extract requests to the UIM after they have been deleted from the EXTLIB by doing the following:

1. Update the EXTLIB by submitting at one time all of those extract requests you run on a regular basis.
2. Create a backup copy of the EXTLIB data set on tape or disk using the Access Methods Services (AMS) REPRO command.
3. Run the DEM, and in the same job delete the EXTLIB and recreate it using the AMS REPRO command and the backup copy of the EXTLIB as input.
Chapter 14. Relational Data Extract feature

This chapter provides information on administering the Relational Data Extract feature, and information on protecting your relational data sources. It assumes that you have the DataRefresher MVS/ESA or VM/ESA base product and the appropriate Relational Data Extract feature installed at your site.

Task overview

The administrative tasks for the Relational Extract Manager are:

1. **Implement a data protection scheme.**
   
   You can use the following to protect your relational data sources:
   
   - Relational database views (DB2 and SQL/DS)
   - SQL authorization statements (DB2 and SQL/DS)
   - CONNECT authority (SQL/DS only)

   For information on maintaining data security while using the Relational Data Extract feature, see "Implementing a data-protection scheme for the Relational Data Extract feature" on page 146.

2. **Edit DataRefresher models to suit your site’s requirements.**

   You can edit the extract request, JCL, and JCS models that DataRefresher provides and make these available to your users. Doing so can save your DataRefresher users time and help reduce the chance for errors. For information and examples of how to edit these models, see *DataRefresher MVS and VM User's Guide*.

3. **Administer the DataRefresher online commands (MVS only; optional).**

   If you choose to use DataRefresher in the online environment, you can tailor that environment by changing the list of DataRefresher online commands keyword defaults. This will simplify and shorten keyword input.

   For information on setting up the DataRefresher online commands for use at your site, see Chapter 15, “Setting up online commands” on page 149.

4. **Administer the DataRefresher dialogs (optional).**

   If you use DataRefresher dialogs at your site, you can tailor each user's dialogs environment, and thus simplify DataRefresher operation for your users.

   For information on starting and using DataRefresher dialogs, see:
   
   - Chapter 16, “Setting up MVS dialogs” on page 157
   - Chapter 17, “Setting up VM dialogs” on page 165
   - *DataRefresher MVS and VM User's Guide*
Implementing a data-protection scheme for the Relational Data Extract feature

This section provides information on protecting your relational data sources from unauthorized use while extracting data with the Relational Data Extract feature. You can use the following methods to maintain the security of your DataRefresher data and resources:

- Relational database views (DB2 and SQL/DS)
- SQL authorization statements (DB2 and SQL/DS)
- CONNECT authority (SQL/DS only)

For full information on DB2 security, authorization considerations, and views, see *IBM DATABASE 2 Version 2: Database Administration Guide*.

For full information on SQL/DS security, authorization considerations, and views, see *SQL/Data System Database Administration for IBM VM Systems*.

Controlling access to data with relational database views

Users do not always need or want to see all the data in a table, and sometimes they may want to combine data from two or more tables. DB2 and SQL/DS use relational database views to provide users with flexibility in the way they look at data.

Relational database views are crucial to controlling access to sensitive data. As an administrator, you can easily define a view that presents only certain rows or columns in a table. Then you can grant appropriate authorization for the view, rather than for the underlying table. You can define views and grant and revoke authorization to access the views to protect your site's sensitive data from unauthorized and inadvertent data access.

A relational database view is an alternative way of looking at the data in tables. It is like a window through which users can access just the data they want. A view is a flexible structure and can present any or all of the data in one or more tables. Views are easy to define and can, in most cases, be used interchangeably with tables.

Creating a view does not store additional copies of the data; rather, the view shows data current at the moment of its use. Views are automatically updated when the administrator updates the tables upon which they are based.

*Note:* Relational database views are not the same as DXTVIEWs. A relational database view is created by DB2 or SQL/DS and is related only to relational databases. DXTVIEWs are related to data descriptions for non-relational databases and files, and are created by DataRefresher.

Granting and revoking data access with SQL authorization statements

As a database administrator, you can use SQL authorization statements to grant and revoke authority for all levels of data and program access in DB2 or SQL/DS at your site. Using these statements, administrators or data owners (users who have defined a table) can limit user access. For example, you can specify that a particular user can update only certain columns in a table. Additionally, you can authorize the use of SQL statements. Thus, by using SQL authorization
statements, you can ensure that only properly authorized users perform database operations at your site.

Both DB2 and SQL/DS also provide your site with the capability to either centralize or decentralize the various authorities that protect data. In a decentralized location, some users may be granted authority over a subset of data, and they, in turn, may grant others access to that data.

**Using CONNECT authority to control access to an SQL/DS database**

Because any user machine can initiate communication with the database machine, SQL/DS must verify that a given user is authorized to access data with that machine. The authority to use a particular SQL/DS database is called CONNECT authority.

A user must have CONNECT authority for a given database before SQL/DS lets that user perform any database functions, such as SQL commands. A user must have CONNECT authority for a database, and each database keeps track of its own authorized users. Thus, if a user needs access to multiple SQL/DS databases, the user must be granted CONNECT authority for each of those databases.

To control access to the data managed by SQL/DS, you must tell SQL/DS:

- Which users are authorized to use the database and protect data from unauthorized access via passwords.
- When a particular user wants to begin using the database.

The SQL/DS database administrator assigns each database user:

- An SQL/DS user ID that is unique for the database.
- A password that is not necessarily unique in the database. (The password is optional.)

Once a user receives CONNECT authority (assigned an SQL/DS user ID), the user can begin to use SQL/DS to access the database via the connect function.

Users can explicitly connect to the SQL/DS database by specifying a user ID and password in their extract requests in the VMCONN keyword of the SUBMIT command. The specified user ID must be in the SQL/DS system catalog, SYSTEM.SYSUSERAUTH.

Users can implicitly connect to SQL/DS by not specifying the VMCONN keyword on a SUBMIT command. That is, in this case, DataRefresher connects to SQL/DS by using the user ID on which the Relational Extract Manager is running.

For more information about connecting to SQL/DS, see *SQL/Data System Database Administration for IBM VM Systems.*
Chapter 15. Setting up online commands

This chapter discusses the general steps you must take to set up the DataRefresher online commands on an MVS host system to meet the needs of your DataRefresher users.

Task overview

To use DataRefresher online commands at your site, you must:

1. Ensure you have the necessary DataRefresher features installed.
   For information on the relationship between DataRefresher online commands and the DataRefresher features, see “Using DataRefresher online commands with DataRefresher features.”

2. Set up your environment to point to the DataRefresher online commands library.
   For information on where the DataRefresher online commands will reside and how they can be called after installation, see “Setting up the online DataRefresher environment” on page 150.

3. Create the required libraries.
   For information on how to create DataRefresher online commands libraries, see “Creating the required libraries” on page 150.

4. Create and/or modify default parameters.
   For information on setting DataRefresher online commands defaults, see “Creating or modifying the defaults file” on page 152.

5. Tailor the DataRefresher online commands to suit your user’s requirements.
   For information on customizing the DataRefresher online commands for use at your site, see “Tailoring DataRefresher online commands for your users” on page 155.

Using DataRefresher online commands with DataRefresher features

Because the DataRefresher online commands call the UIM, DEM, or REM, the corresponding DataRefresher feature must be installed and administered to make the required program available. The UIM and DEM are parts of the General Data Extract feature. The REM is part of the Relational Data Extract feature.
The DataRefresher online commands run the following programs:

<table>
<thead>
<tr>
<th>Online command</th>
<th>Program executed</th>
<th>Command generated</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCANCEL</td>
<td>UIM</td>
<td>CANCEL</td>
</tr>
<tr>
<td>DCREATE</td>
<td>UIM</td>
<td>CREATE</td>
</tr>
<tr>
<td>DDELETE</td>
<td>UIM</td>
<td>DELETE</td>
</tr>
<tr>
<td>DLIST</td>
<td>UIM</td>
<td>LIST</td>
</tr>
<tr>
<td>DPRINT</td>
<td>UIM</td>
<td>PRINT</td>
</tr>
<tr>
<td>DPUNCH</td>
<td>UIM</td>
<td>PUNCH</td>
</tr>
<tr>
<td>DSTATUS</td>
<td>UIM</td>
<td>STATUS</td>
</tr>
<tr>
<td>DSEND</td>
<td>UIM</td>
<td>SUBMIT/EXTRACT (UIM)</td>
</tr>
<tr>
<td>DRUN</td>
<td>DEM</td>
<td>INITDEM/USE</td>
</tr>
<tr>
<td>DRUNR</td>
<td>REM</td>
<td>SUBMIT/EXTRACT (REM)</td>
</tr>
</tbody>
</table>

Setting up the online DataRefresher environment

After DataRefresher has been installed at your site, the DataRefresher online commands will reside in the DVR110.DVRSAMPE library. These REXX EXEs can be kept in a CLIST or REXX EXEC library. For example, you can move them to DVR110.REXX.EXEC, using jobs discussed in the appropriate DataRefresher program directory.

Allocate the REXX EXEs to SYSPROC to allow the DataRefresher online commands to be called without pointing to a command library. For example:

ALLOC FILE(SYSPROC) DATASET('DVR110.REXX.EXEC' - 'SYS1.TS0P.CLIST') SHR REUSE

This example concatenates your REXX EXEs to other CLIST libraries allowing you to call the DataRefresher online commands at the TSO command line. Add a similar allocation to your TSO initialization.

Creating the required libraries

To create the required DataRefresher online commands data sets, run the REXX EXEC called DXTALLOC. It will create the required DataRefresher libraries for command input, output, JCS, output listings, and dumps.
For example, to create all Online DataRefresher data sets with your TSO user ID, enter DXTALLOC at a TSO command line.

Or, to create all DataRefresher Online data sets with the prefix MYUSERID, enter: DXTALLOC PREFIX=MYUSERID

This command creates the following data sets:

**myuserid.DXTINOUT**

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
<th>Allocated to ddname</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCREATE</td>
<td>DCREATE input data</td>
<td>DXTIN</td>
</tr>
<tr>
<td>DSEND</td>
<td>DSEND input data</td>
<td></td>
</tr>
<tr>
<td>DRUNR</td>
<td>DRUNR input data</td>
<td></td>
</tr>
<tr>
<td>DRUN</td>
<td>DRUN input data set</td>
<td></td>
</tr>
<tr>
<td>DRUNO</td>
<td>DRUN output data</td>
<td>DXTOUT</td>
</tr>
<tr>
<td>DRUNRO</td>
<td>DRUNR output data</td>
<td></td>
</tr>
<tr>
<td>DPUNCH</td>
<td>DPUNCH output data</td>
<td>DXTPUNCH</td>
</tr>
<tr>
<td>DSENDJ</td>
<td>DSEND JCS</td>
<td>DXTJCS</td>
</tr>
<tr>
<td>DRUNRJ</td>
<td>DRUNR JCS</td>
<td></td>
</tr>
<tr>
<td>DEFAULTS</td>
<td>Defaults data</td>
<td>DEFAULTS</td>
</tr>
</tbody>
</table>

**myuserid.DXTPRINT**

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
<th>Allocated to ddname</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None)</td>
<td>Sequential data set for output listing</td>
<td>DXTPRINT</td>
</tr>
</tbody>
</table>

**myuserid.DXTDIAG**

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
<th>Allocated to ddname</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None)</td>
<td>Sequential data set for DFSORT messages</td>
<td>SYSOUT</td>
</tr>
</tbody>
</table>

**myuserid.DXTDUMP**

<table>
<thead>
<tr>
<th>Members</th>
<th>Description</th>
<th>Allocated to ddname</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None)</td>
<td>Sequential data set for dump listing</td>
<td>DXTDUMP</td>
</tr>
</tbody>
</table>

If you do not run DataRefresher requests with a DEBUG value greater than 1, you can delete the DXTDUMP data set. A DEBUG value greater than 1 is used to test exit routines you may write, or to create diagnostic information for problem reporting to IBM.
The allocation parameters for these data sets are:

```plaintext
myuserid.DXTINOUT:
  DSN(dxtinout) DD(dxtin) BLK(800) LRECL(80)
  RECFM(f,b) SPACE(200,20) DSORG(po) DIR(5)
  NEW CATALOG REUSE

myuserid.DXTPRINT:
  DSN(dxtprint) DD(dxtprint) BLK(1210) LRECL(121)
  RECFM(f,b) SPACE(150,20) DSORG(ps)
  NEW CATALOG REUSE

myuserid.DXTDIAG:
  DSN(dxtdiag) DD(sysout) BLK(1210) LRECL(121)
  RECFM(f,b) SPACE(25,1) DSORG(ps)
  NEW CATALOG REUSE

myuserid.DXTDUMP:
  DSN(dxtdump) DD(dxtdump) BLK(3146) LRECL(121)
  RECFM(f,b) SPACE(25,5) DSORG(ps)
  NEW CATALOG REUSE
```

Change these space allocations as needed to suit your installation requirements. DXTALLOC is run once, to catalog the DataRefresher online commands data sets for a user ID. Thereafter, each Online DataRefresher command allocates these data sets each time the DataRefresher online commands are used. You should run DXTALLOC for every user ID using the DataRefresher online commands.

---

**Creating or modifying the defaults file**

The online DataRefresher defaults file is built by DXTALLOC as `prefix.DXTINOUT(DEFAULTS)`. The online REXX EXECs reference the defaults file using that name. If this name is changed, you will have to change the name in each EXEC that appears as `DEFAULTS = prefix.DXTINOUT(DEFAULTS)`.

The member called DEFAULTS represents a list of DataRefresher default parameters used by the DataRefresher online REXX EXECs. The list can be edited to reflect installation standards which minimize command line input.
When using the General Data Extract feature, for example, if all DXTFILE descriptions are going to reside in the FDTLIB named MYNAME.FDTLIB2, then you can edit DEFAULTS, changing:

```
FDTL = DVR110.FDTLIB    /* FDTLIB NAME */
```
to:

```
FDTL = MYNAME.FDTLIB2    /* FDTLIB NAME */
```

Otherwise you can make the same change at a TSO command line, when calling an Online DataRefresher command that has the FDTL optional keyword. For example, to print something from the FDTLIB using the DPRINT command, type:

```
DPRINT ....,FDTL=MYNAME.FDTLIB2
```

where FDTL=MYNAME.FDTLIB2 is used to override the defaults file FDTLIB setting.

Similarly, you can modify any entry in the defaults file. See the DataRefresher Command Reference for the complete descriptions of each of these keywords.

The other keyword values you can set in the defaults file include:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Specifies</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTL</td>
<td>The EXTLIB data set name.</td>
</tr>
<tr>
<td>LOADLIB</td>
<td>The DXT load library to be used for this online program.</td>
</tr>
<tr>
<td>REQID</td>
<td>The extract identifier of a specific extract for the DRUN command to run.</td>
</tr>
<tr>
<td>INPUT</td>
<td>Whether or not to show EDIT screen of DXTIN. Used for DCREATE, DRUNR, and DSEND commands.</td>
</tr>
<tr>
<td>JCS</td>
<td>Whether or not JCS is used and shown on EDIT screen. Used for DRUNR and DSEND commands. If you are using JCS this should be either YES or EDIT. The default is initially NO.</td>
</tr>
<tr>
<td>BROWSE</td>
<td>Whether or not to browse the DXTPRINT data set.</td>
</tr>
<tr>
<td>PRINT</td>
<td>Whether or not to print the DXTPRINT data set.</td>
</tr>
<tr>
<td>PREFIX</td>
<td>The high-level qualifier of the DataRefresher required data sets that were built by DXTALLOC. Only needed if this is different from your TSO user ID.</td>
</tr>
<tr>
<td>MODEL</td>
<td>The model you want copied into the DXTIN data set. Used for DCREATE, DSEND, and DRUNR commands.</td>
</tr>
<tr>
<td>MODELDS</td>
<td>The data set containing the models used for DCREATE, DSEND, and DRUNR commands.</td>
</tr>
<tr>
<td>JCSDD</td>
<td>The ddname coded on the JCS keyword of the SUBMIT command (if specified). Used for DSEND and DRUNR commands, when JCS=YES or JCS=EDIT was specified.</td>
</tr>
<tr>
<td>JMODEL</td>
<td>The JCS model to be used. Used for DSEND command.</td>
</tr>
<tr>
<td>LANG1LIB</td>
<td>The language library used for exit routines. Used when EXITLANG is specified.</td>
</tr>
</tbody>
</table>
LANG2LIB The second language library used for exit routines. Used when EXTLANG is specified and two language libraries need to be used.

EXITLANG The languages used for exit routines.

EXITLIB The load library for exit routines if it is not the load library containing DataRefresher.

EXTDATA Whether or not DataRefresher writes the output to an external data set. Code YES only if EXTDATA=ddname will be coded on the SUBMIT command. The Online command will then allocate the sequential data set specified by the ddname on the EXTDD keyword, and the data set name specified on the EXTDS keyword. If EXTDATA=dsnname will be coded on the SUBMIT command, specify EXTDATA=N0. DataRefresher dynamically allocates the data set.

EXTDD The ddname specified in the SUBMIT command with the EXTDATA keyword.

EXTDS The data set name corresponding to the EXTDD ddname.

DXTFILE The name of a DXTFILE used in the extract request.

DXT2FILE The name of a DXTFILE used in the extract request if one or more DXTFILEs are being joined.

DXT3FILE The name of a DXTFILE used in the extract request if one or more DXTFILEs are being joined.

DS1NAME The name of the data set associated with DXT1FILE.

DS2NAME The name of the data set associated with DXT2FILE.

DS3NAME The name of the data set associated with DXT3FILE.

PLAN The DB2 plan used by the REM when called by DRUNR.

DB2ID The database name used by the REM when called by DRUNR.

DB2LOAD The library containing the executable DB2 code.

PUSER The user ID to route DXTPRINT output to when PRINT=YES is specified. This keyword can only be set in the defaults file, not on a command line.

PDEST The destination node ID to route output to when PRINT=YES is specified. This keyword can only be set in the defaults file, not on a command line.

The following keywords are specified within the UIM options (OPTION keyword):

DEBUG The debugging level.

EXITLANG The language environments required by exit routines.

MIXED Whether the WHERE keyword of the SUBMIT command contains mixed string of DBCS and single-byte characters.

GDI GDI user parameters passed to the exit routine.
Tailoring DataRefresher online commands for your users

You can tailor the DataRefresher online commands to suit the requirements of the DataRefresher users at your site. Consider performing the following tasks:

- Set up your defaults file.
  
  You can set frequently used optional keywords in the defaults file. For example:
  - Set up the default EXTLIB, FDTLIB, and load libraries.
  - To use specific models, set the MODEL keyword to the name of a frequently used input model and the JMODEL keyword to the name of a frequently used JCS model.
  - To set your printing preference, set the PRINT keyword to YES if you want your output printed, NO if a browse at the terminal is enough, or PROMPT if you will be doing both. Then set the PUSER and PDEST keyword in the defaults file to the destination ID and node for the print output.

Set these defaults to handle the majority of commands, in order to limit the command line keywords needed.

- Modify DataRefresher models to suit your requirements.

  Because the DataRefresher models are very general, you can edit them to be specific to their common use at your site. For example, suppose the CREATE DXTVIEW model (DVREDRVF) will be frequently used. You can remove all of the ending comments and fill in an example of a common DXTVIEW. Then you can store this edited version with a different name and retain the original model intact.

- Use CLISTs to group the DataRefresher online commands to automate their use.

  For example, suppose you want to create and run repeatedly an extract with the extract ID MYEXT1 using the DXTFILE DATA1 and the data set name MYID.DATA1. Assuming the defaults file is set with the correct JCS values, and your SUBMIT/EXTRACT command is in member MYSUBMIT in the default model data set, the CLIST could look like this:

  ```
  PROC 0
  DSEND INPUT=NOEDIT,MODEL=MYSUBMIT,PRINT=YES
  DRUN REQID=MYEXT1,DXTFILE=DATA1,DSNAME=MYID.DATA1,PRINT=YES
  EXIT
  ```

  If you name this CLIST MYEXT1, then you can run MYEXT1 when you want to extract this data, and print the DXTPRINT data set.

Where to look for more information

For more information on using DataRefresher online commands, see the *DataRefresher MVS and VM User's Guide*. If you know how to use DataRefresher online commands to create and maintain data descriptions and extract requests, but need to refer to the syntax diagrams and keyword explanations of these commands, see the *DataRefresher Command Reference*.  

Chapter 15. Setting up online commands  155
Chapter 16. Setting up MVS dialogs

This chapter describes how to prepare for using the DataRefresher Administrative Dialogs and End User Dialogs in the MVS operating environment. For full information on starting using the DataRefresher dialogs, see the DataRefresher MVS and VM User's Guide.

Task overview

To prepare to use the DataRefresher dialogs at your site, you must:

1. Save or rename any dialog user data sets you have from DXT Version 2 Releases 4 or 5 (optional).
   For information on using dialogs data sets created for use with DXT, see “Using tables and data sets from DXT dialogs.”

2. Edit MVS CLISTs for your environment.
   For information on editing DVRELNRL and DVRELLGN CLISTs for use at your site, see “Editing CLISTs for your environment” on page 158.

3. Enroll the administrator.
   For information on calling DVRELNRL CLIST to set up your DataRefresher libraries, see “Enrolling the administrator” on page 161.

4. Edit DataRefresher dialogs models to suit your site's requirements.
   For information on locating and working with models of data descriptions, extract requests, JCL, JCS and SAP skeletons, see “DataRefresher dialogs models” on page 162.

5. Maintain library space for your users.
   For information on ensuring that you have enough library space for DataRefresher users at your site, see “Maintaining library space for your users” on page 164.

6. Administer the End User Dialogs.
   For task overviews covering the steps necessary to administer the End User Dialogs, see “Administering the End User Dialogs” on page 164.

Using tables and data sets from DXT dialogs

This section provides information you need to use dialogs tables and data sets from DXT Version 2 Releases 4 and 5.

DXT dialog components that are compatible with DataRefresher

The following ISPF tables from DXT Version 2 Releases 4 and 5 are compatible with DataRefresher Version 1, and can be used without change:

- JCL table (DVRRJCLN)
- Administrative Dialogs extract request table (DVRREXRQ)
- End User Dialogs extract request table (DVRRPXRQ)
- Data description table (DVRRDDRQ)
- Dialogs User Profile (DVRDPROF)
Renaming DXT dialog user data sets

If you want to use your user data sets from DXT Version 2 Releases 4 or 5, you can:

- Copy and save existing data sets
  If you want to save your data sets, copy and rename them according to the DataRefresher Version 1 naming conventions shown in Table 26.
- Rename existing data sets
  If you want to use the same data sets, rename them according to the DataRefresher Version 1 naming conventions shown in Table 26.

<table>
<thead>
<tr>
<th>DXT Version 2 Release 4</th>
<th>DXT Version 2 Release 5</th>
<th>DataRefresher Version 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXT240.DVRTLIBE</td>
<td>DXT250.DVRTLIBE</td>
<td>DVR110.DVRTLIBE</td>
</tr>
<tr>
<td>DXT240.DVRIMEXE</td>
<td>DXT250.DVRIMEXE</td>
<td>DVR110.DVRIMEXE</td>
</tr>
<tr>
<td>DXT240.DVRFJEDIE</td>
<td>DXT250.DVRFJEDIE</td>
<td>DVR110.DVRFJEDIE</td>
</tr>
<tr>
<td>DXT240.DVRDPROF</td>
<td>DXT250.DVRDPROF</td>
<td>DVR110.DVRDPROF</td>
</tr>
</tbody>
</table>

The user ID, normally appended to the data set names as a high-level qualifier, is not shown in Table 26.

Using an ISPF profile data set (ISPPROF)

The DataRefresher Version 1 profile data sets are named DVRDPROF. If you were an ISPF user before becoming a DXT or DataRefresher dialogs user, you might still be using your ISPF profile data set (ISPPROF). If so, you do not need to create a DataRefresher profile data set.

Editing CLISTs for your environment

DataRefresher dialogs have several CLISTs you can use to set up DataRefresher. You use these CLISTs to set up dialogs libraries and enroll a user.

You must edit the following CLISTS to include installation-specific information:

**DVRELNRL** Creates and initializes the user specific data sets required by each dialog user.

**DVRELLGN** Establishes access to the data sets needed for the dialogs (allocates them). DVRELNRL must have already been run. The user automatically calls this CLIST from the ISPF main menu when choosing a DataRefresher dialogs option.

These CLISTs are distributed in DVR110.DVRSAMPE. DVRELNRL can be edited and run from your DataRefresher library, but DVRELLGN must be in your system CLIST library.
Editing DVRELNRL

DVRELNRL is run from the administrator's user ID. It performs these specific functions:

- Prompts the administrator for the user ID of the user to be enrolled.
- When enrolling the administrator, copies the JCL and request models provided with DataRefresher dialogs from the DVR110.DVRJEDIE into the &userid.DVR110.DVRJEDIE data set.
- When enrolling a user other than the administrator, copies:
  - The JCL and request models from the administrator's &userid.DVR110.DVRJEDIE into the user's &userid.DVR110.DVRJEDIE data set.
  - DVRDPROF from the administrator's &userid.DVR110.DVRDPROF to the new user's profile data set.
- When enrolling a new user, creates libraries DVRTLIBE, DVRDPROF, and DVRIMEXE.

Make the following changes to DVRELNRL, if necessary:

1. Change the following parameters on the PROC statement to conform to your site's requirements using the ISPF editor:

   DXTPRE    The DataRefresher data set prefix.
   DXUNIT    The unit that has the DataRefresher production code.
   DXTVOL    The volume that has the DataRefresher production code.
   ADMINID   The user ID of the DataRefresher administrator. If this is the same as the user ID being enrolled, DataRefresher dialogs will set up this user as the administrator. For other users, this serves as the user ID prefix on data sets from which the CLIST copies the needed members.
   DSN       The parameter for indicating the name of your JCL and edit request data set, if the name is other than &userid.&DXTPRE..DVRJEDIE.

2. If you are enrolling a user who is already a DataRefresher user, delete the section of DVRELNRL that creates DVRTLIBE, DVRDPROF, and DVRIMEXE. Substitute the name of the established profile data set on the COPY statement.

3. Change the storage allocations, if necessary.

If non-default names were entered during the installation process, change the default names listed in the CLIST to the new names. The default names are:

- DVRIMEXE
- DVRDPROF
- DVRJEDIE

Appendix E, "DataRefresher dialogs data set and file definitions" on page 203 contains a description of these files.
Editing DVRELLGN

DVRELLGN is called when a user selects to run either the End User Dialogs or the Administrative Dialogs. ISPF must be called first, from the logon procedure the user runs when initiating a session. (Run this logon procedure before you call a DataRefresher dialogs session. The CLIST sets up the ISPF environment and calls ISPF.)

The DVRELLGN CLIST:

- Allocates all the files that DataRefresher dialogs need for execution, using ISPF LIBDEF functions
- Prompts the user to specify whether object sharing is in effect during this session
- Prompts the user to specify whether DataRefresher sample data will be used during this session

To conform to your site's requirements, change the following parameters on the PROC statement using the ISPF editor:

**OPTION** Indicates the type of DataRefresher dialogs session. It is passed as a D if the Administrative Dialogs are selected, or as a blank or a valid send command if the End User Dialogs are selected. This parameter should be specified by the user on the ISPF main menu when the CLIST is called.

**DXTPRE** The DataRefresher data set prefix.

**DSN** Indicates the name of your request data set, if the name is other than &userid..&DXTPRE..DVRIJEDIE.

**SHARE** Indicates whether object sharing will be used. It is passed as a Y if the user intends to use object sharing, or an N if the user does not intend to use object sharing during this session.

If left blank, the CLIST prompts the user to determine whether shared objects will be used. If you specify a Y or an N when the CLIST is called from the ISPF main menu, the user does not have the option to choose object sharing. See the DataRefresher MVS and VM User's Guide for more information about object sharing.

**SAMPLE** Indicates whether DataRefresher sample data will or will not be used. It is passed as a Y if the user only wants to use DataRefresher sample data, or an N if the user does not want to use DataRefresher sample data. If left blank, the CLIST prompts the user to specify whether DataRefresher sample data will be used.

If non-default names were entered during the installation process, change the default names listed in the CLIST to the new names. The default names are:

- DVRPLIBE
- DVRMLIBE
- DVRSLIBE
- DVRADME
- DVRTLIBE
- DVRLDAD
- DVLREUSE
Appendix E, "DataRefresher dialogs data set and file definitions" on page 203 contains a description of these data sets.

Setting up the end user environment

To set up an End User Dialogs environment so that a user:

- Always gets object sharing
- Uses a JCL/Edit library named THIMB.DVR110.THIMBREQ
- Does not use DataRefresher sample data

code the ISPF main menu PROC section as shown in Figure 40.

```plaintext
)PROC
&ZSEL = TRANS( TRUNC (&ZCMD, '.')
0,'PANEL(ISOPTA)'
1,...
E,'CMD(%DVREXUSR SHARE(Y) (DSN)(THIMB.DVR110.THIMBREQ) SAMPLE(N)'
```

Figure 40. End User Dialogs setup

End of Product Sensitive Programming Interface

Enrolling the administrator

This section provides information on how to run the DVRELNRL CLIST, which sets up a user’s DataRefresher library in the MVS environment. Before you can run DVRELNRL, you must edit it. For information on editing DVRELNRL, see “Editing CLISTs for your environment” on page 158.

Calling the DVRELNRL CLIST

Use the following steps to call the DVRELNRL CLIST:

1. Log on to the administrator’s user ID.

2. If you pass PROC parameters to the CLIST during invocation, refer to the parameters given in step 3 as a reference. Otherwise, run the CLIST and start your DataRefresher dialogs session.

3. Use these parameters on the PROC statement:

   - **DXTPRE** The DataRefresher data set prefix.
   - **DXUNIT** The unit that has the DataRefresher production code.
   - **DXTVOVOL** The volume that has the DataRefresher production code.
   - **ADMINID** The user ID of the administrator that controls the model requests and JCL files to be used as a basis for the user's &userid.&DXTPRE.DVRJEDIE, and the profile member to be copied into &userid.&DXTPRE.DVRDPROF.
   - **DSN** Identifies the name of your request data set, if the name is other than &userid.&DXTPRE..DVRJEDIE.

4. Run the CLIST.
How the CLIST runs

If you are enrolling:

- The DataRefresher administrator, enter the administrator's user ID and press ENTER. If you do not want to enroll other users, press ENTER again in response to the prompt.
- Users other than the administrator, the CLIST prompts you for their user IDs. For each user you want to enroll, enter the user ID and press ENTER. When you have enrolled all users, press ENTER in response to the prompt.

When the CLIST is complete, a DataRefresher dialogs library should have been built for each user being enrolled.

For information on using DataRefresher dialogs, see the DataRefresher MVS and VM User's Guide.

DataRefresher dialogs models

DataRefresher models provide you with examples of how to complete a number of DataRefresher tasks. You can find the models in a data set called DVR110.DVRJEDIE. If you would like to see a model, you can:

- Call the model up during a DataRefresher dialogs session
- Browse the model outside the dialogs, as you would any other data set to which you have access
- Print the model using the procedure you have to print a data set
- Browse and edit the model while using DataRefresher online commands

Each of the models contains instructions to help you edit it to suit your site's requirements.

Data description models

Table 27 describes the data description models provided with DataRefresher. These models are located in the data set DVR110.DVRJEDIE.

<table>
<thead>
<tr>
<th>MODEL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVREDREF</td>
<td>Creates a DXTFILE description</td>
</tr>
<tr>
<td>DVREDREP</td>
<td>Creates a DXTPSB description</td>
</tr>
<tr>
<td>DVREDCRD</td>
<td>Creates a DATATYPE description</td>
</tr>
<tr>
<td>DVREDRVF</td>
<td>Creates a DXTVIEW of a DXTFILE description</td>
</tr>
<tr>
<td>DVREDRVVP</td>
<td>Creates a DXTVIEW of a DXTPCB description</td>
</tr>
</tbody>
</table>
Extract request models

Table 28 describes the extract request models provided with DataRefresher. These models are located in the data set DVR110.DVRJEDIE.

<table>
<thead>
<tr>
<th>MODEL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVREDRXM</td>
<td>Submits REM extract requests on MVS using the SUBMIT(REM) command</td>
</tr>
<tr>
<td>DVREDEXT</td>
<td>Submits DEM extract requests using the SUBMIT(UIM) command</td>
</tr>
</tbody>
</table>

JCL and JCS models

Table 29 describes the JCL and JCS models provided with DataRefresher. These models are located in the data set DVR110.DVRJEDIE.

<table>
<thead>
<tr>
<th>MODEL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVREDREM</td>
<td>Starts the REM when extracting from a DB2 source</td>
</tr>
<tr>
<td>DVREDDAP</td>
<td>Starts the DAP</td>
</tr>
<tr>
<td>DVREDDXT</td>
<td>Starts the DXTINPUT procedure</td>
</tr>
<tr>
<td>DVREDDJC</td>
<td>Loads extracted data into DB2</td>
</tr>
<tr>
<td>DVREDBC</td>
<td>Creates and loads into DB2</td>
</tr>
<tr>
<td>DVREDSJC</td>
<td>Loads extracted data into SQL/DS</td>
</tr>
<tr>
<td>DVREDRUM</td>
<td>Executes the DRU in MVS</td>
</tr>
<tr>
<td>DVREDEJM</td>
<td>Executes DataRefresher's MIT utility in MVS</td>
</tr>
</tbody>
</table>

SAP skeletons

Table 30 describes the SAP skeleton models provided with DataRefresher. These models are located in the data set DVR110.DVRJEDIE.

<table>
<thead>
<tr>
<th>NAME</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVRXKJDA</td>
<td>SAP dialog (CREATE DXTPSB)</td>
</tr>
<tr>
<td>DVRXKJFA</td>
<td>SAP Dialog (CREATE DXTFILE)</td>
</tr>
</tbody>
</table>
Maintaining library space for your users

As administrator, you must maintain library space allocations for the DataRefresher users at your site. If there is insufficient space available, users may receive a warning while using a DataRefresher dialogs session.

You maintain library space by doing either of the following:

- Using the COMPRESS function of the ISPF library utility to reorganize the space in an existing data set.
- Using the ISPF library utilities DATASET (allocate) to create a new or larger data set; and DATASET (copy) to copy the contents from an existing library to a new library. You also use DATASET to delete the previous data set and rename the new one.

Administering the End User Dialogs

Following are task overviews for building a nickname table, updating an end user table, and requesting source descriptions (updating a master index table).

For full DataRefresher dialogs operating procedures, see the *DataRefresher MVS and VM User's Guide*.

Building the nickname table

1. Log on to Administrative Dialogs
2. From the main menu, select option 5 to display the END USER DIALO GS ADMINISTRATION panel
3. Select option 1 to display the ESTABLISH NICKNAMES FOR SYSTEMS panel
4. Specify information for nicknames

Updating the end user table

1. Log on to Administrative Dialogs
2. From the main menu, select option 5 to display the END USER DIALO GS ADMINISTRATION panel
3. Select option 2 to display the UPDATE THE END USER TABLE panel
4. Fill in appropriate source node ID, source user ID, and DataRefresher dialogs user ID

Requesting source descriptions

1. Log on to Administrative Dialogs
2. From the main menu, select option 5 to display the END USER DIALO GS ADMINISTRATION panel
3. Select option 3 to display the REQUEST SOURCE TABLE DESCRIPTION(S) panel
4. Fill in fields on REQUEST SOURCE TABLE DESCRIPTION(S) panel
5. Specify tables or DataRefresher views you want
6. Review JCL and Load JCS, if necessary
7. Review job stream if necessary
8. Submit job to UIM or REM
Chapter 17. Setting up VM dialogs

This chapter describes how to prepare to use the DataRefresher Administrative Dialogs and End User Dialogs in the VM operating environment. Refer to the DataRefresher MVS and VM User's Guide for further information on DataRefresher dialogs.

Task overview

To prepare to use the DataRefresher dialogs at your site, you must:

1. **Save any MACLIBs you have from DXT Version 2 Releases 4 or 5 (optional).**
   
   For information on using MACLIBs created for use with DXT, see “Using tables and MACLIBs from DXT dialogs.”

2. **Edit VM EXECs for your environment.**
   
   For further information, see “Editing VM EXECs for your environment” on page 166.

3. **Enroll the administrator.**
   
   For information on calling the DVREXLIB EXEC to set up your DataRefresher libraries, see “Enrolling the administrator” on page 169.

4. **Edit DataRefresher dialogs models to suit your site's requirements.**
   
   For information on locating and working with models of data descriptions, extract requests, JCL, JCS and SAP skeletons, see “DataRefresher dialogs models” on page 170.

5. **Maintain library space for your users.**
   
   For information on ensuring that you have enough library space for DataRefresher users at your site, see “Maintaining library space for your users” on page 172.

6. **Administer the End User Dialogs.**
   
   For task overviews covering the steps necessary to administer the End User Dialogs, see “Administering the End User Dialogs” on page 172.

Using tables and MACLIBs from DXT dialogs

This section provides information you need to use dialogs tables and MACLIBs from DXT Version 2 Releases 4 and 5.

DXT dialog components that are compatible with DataRefresher

The following ISPF tables from DXT Version 2 Releases 4 and 5 are compatible with DataRefresher Version 1, and can be used without change:

- JCL table (DVRRJCLN)
- Administrative Dialogs extract request table (DVRRERXQ)
- End User Dialogs extract request table (DVRRPXRQ)
- Data description table (DVRRDDRO)
- Dialogs User Profile (DVRRDPROF)
Using DXT dialog user MACLIBs

If you want to use your user MACLIBs from DXT Version 2 Releases 4 or 5, you can:

- Copy and save existing MACLIBs

  If you want to save your MACLIBs, copy and rename them differently from the DataRefresher Version 1 naming conventions shown in Table 31.

- Use existing MACLIBs

  If you want to use the same MACLIBs with DataRefresher Version 1, you do not have to rename them.

Table 31. Default names for DXT and DataRefresher dialogs MACLIBs

<table>
<thead>
<tr>
<th>DXT Version 2 Release 4</th>
<th>DXT Version 2 Release 5</th>
<th>DataRefresher Version 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVRTLIBE MACLIB</td>
<td>DVRTLIBE MACLIB</td>
<td>DVRTLIBE MACLIB</td>
</tr>
<tr>
<td>DVRIMEXE MACLIB</td>
<td>DVRIMEXE MACLIB</td>
<td>DVRIMEXE MACLIB</td>
</tr>
<tr>
<td>DVRJEDIE MACLIB</td>
<td>DVRJEDIE MACLIB</td>
<td>DVRJEDIE MACLIB</td>
</tr>
<tr>
<td>USERID MACLIB</td>
<td>USERID MACLIB</td>
<td>USERID MACLIB</td>
</tr>
</tbody>
</table>

Using an ISPF profile MACLIB (ISPPROF)

The profile MACLIBs are named USERID MACLIB. If you were an ISPF user before becoming a DataRefresher dialogs user, you might still be using your ISPF profile MACLIB (ISPPROF). If so, you do not need to create a DataRefresher Version 1 profile MACLIB.

Editing VM EXECs for your environment

Product Sensitive Programming Interface

This section explains how to change three EXECs with installation-specific information.

DVREXLIB  Creates and initializes user-specific files for your DataRefresher dialogs users.

DVREXUSR  Allocates the files you need for DataRefresher dialogs. DVREXLIB must have already been run. DVREXUSR is automatically called from the ISPF main menu when choosing a DataRefresher dialogs option.

DVREXSND  Used internally by DataRefresher dialogs, sends requests built during DataRefresher dialogs to the appropriate system.

You use these EXECs to set up a dialogs library, enroll a user, or send a request to a different system (from VM).
Editing DVREXLIB

DVREXLIB runs on the administrator's user ID or on the user ID of the person being enrolled on DataRefresher dialogs. The DVREXLIB EXEC performs these functions in the following order:

1. Identifies the production minidisk created and initialized during the installation of the DataRefresher base product.

2. Prompts the administrator for the user ID of the user to be enrolled.

3. When enrolling a user other than the administrator, prompts the administrator for the filemode of the administrator's disk that contains the DVRJEDIE (JCL and edited requests) and profile to be copied.

4. When enrolling a user other than the administrator, prompts the administrator to specify whether the user to be enrolled was a DXT dialogs user.

5. When enrolling the administrator, copies all models provided with DataRefresher from the DVRJEDIE MACLIB on the production minidisk to a new DVRJEDIE on the administrator's A-disk.

6. When enrolling a user other than the administrator, copies all models provided with DataRefresher from DVRJEDIE MACLIB on the administrator's A-disk to a new DVRJEDIE. It also copies the administrator's profile information into the new user's userid MACLIB.

7. Creates a DVRJEDIE MACLIB for use by the import/export function of DataRefresher dialogs.

8. Sends newly created DVRJEDIE, DVRIMEXE, and the userid MACLIB profile to the user being enrolled.

Change all references to the files on the production minidisk that do not use the default names. If non-default names were entered during the installation process, change the default names listed in the previous EXEC to the new names. The default names are:

- DVRIMEXE
- DVRDPROF
- DVRJEDIE

For a description of these files, see Appendix E, “DataRefresher dialogs data set and file definitions” on page 203.

Editing DVREXUSR

DVREXUSR is called when a user selects to run either the End User Dialogs or the Administrative Dialogs.

The DVREXUSR EXEC:

- Allocates all the files that DataRefresher dialogs need to run, using ISPF LIBDEF and FILEDEF functions
- Prompts the user to specify whether object sharing is in effect during this session
- Prompts the user to specify whether to use DataRefresher sample data during this session.
Check the parameters you pass to the DVREXUSR EXEC. Any changes to these parameters should be made to the PROC area of the ISPF main menu panel:

**OPTION**  This is passed as a 0 if the Administrative Dialogs are selected, or as a blank or valid send command if the End User Dialogs are selected. Specify this parameter on the ISPF main menu when the EXEC is called.

**SHARE**  This is passed as a Y if the user intends to use object sharing, or as an N if the user does not intend to use object sharing during this session. (See the *DataRefresher MVS and VM User's Guide* for more information about object sharing.)

When left blank, the EXEC prompts the user to determine whether to use shared objects. If you specify a Y or an N when the EXEC is called from the ISPF main menu, the user does not have the option to choose object sharing.

**FILENAME**  This is the name of the JCL/Edit objects file. The default name is DVRJEDIE. If you changed the name during installation, you need to either pass the FILENAME parameter or change the EXEC directly to the new name. If you did not change the name during installation, do not change this parameter.

**SAMPLE**  This is passed as a Y only if the user wants to use DataRefresher sample data, or as an N if the user does not want to use DataRefresher sample data. When left blank, the EXEC prompts the user to determine whether to use sample data.

If you plan to use object sharing, the LIBDEF function will be used to allocate DVRTLIBE using the ddname DVRSTABL.

If non-default file names were entered during the installation process, change the default names listed in the previous EXEC to the new names. The default names are: DVRPLIBE, DVRMLIBE, DVRSLIBE, DVRTADME, DVRLOAD, and DVRJEDIE. Appendix E, "DataRefresher dialogs data set and file definitions" on page 203 contains a description of these files.

### Setting up the end user environment

To set up an End User Dialogs environment so that a user:

- Always gets object sharing
- Uses a JCL/Edit library named THIMBREQ MACLIB
- Does not use DataRefresher sample data

code the ISPF main menu PROC section as shown in Figure 41.

```plaintext
)PROC
&ZSEL = TRANS( TRUNC (&ZCMD, '.,')
   0, 'PANEL(ISOPTA)' 1,...
   E,'CMD(DVREXUSR,Y,THIMBREQ,N)'
```

*Figure 41. End User Dialogs setup*
Editing DVREXSND

DVREXSND sends a DataRefresher dialogs request built in the VM environment to the appropriate VM or MVS system. The EXEC is automatically called at the end of the SEND function when you are using the DataRefresher dialogs in the VM environment. Up to four parameters are passed to it:

&1 The name of the generated job is always passed.

&2 The target for extract output is passed when sending a DataRefresher End User Dialogs request or End User Administration request.

&3 The node ID to which a generated job is sent for execution is passed when sending a DataRefresher End User Dialogs request or an End User Dialogs Administration request.

&4 The user ID under which a generated job will run is passed when sending a DataRefresher End User Dialogs request or an End User Dialogs Administration request.

Make the following changes to DVREXSND, if necessary:

1. Provide the name of the RSCS machine where the generated DataRefresher job may be spooled. The default is RSCS.

2. Identify the node where the jobs will be sent. This is the source data system. During an End User Dialogs or End User Dialogs Administration session, the correct node ID is passed to the EXEC by the program. For other DataRefresher dialogs requests, this EXEC needs to be changed to send the request.

3. Name the user ID on which the job will run. During an End User Dialogs or End User Dialogs Administration session, the correct user ID is passed to the EXEC by the program. For other DataRefresher dialogs requests, this EXEC needs to be changed to send the request.

Note: If you have users that are sending edited requests to different user IDs, make sure this EXEC is on each user's A-disk, so that users can change the EXEC. Change the copy on the user's A-disk to include the appropriate information. For more information, see the DataRefresher MVS and VM User's Guide.

4. Specify the node ID from which you send the request for execution.

Once changed, you do not need to change the EXEC again unless you send a request (other than in End User Dialogs and End User Administration) to a system other than the one you have specified on &DATANODE in the EXEC.

| End of Product Sensitive Programming Interface |

Enrolling the administrator

This section provides information on how to run the DVREXLIB EXEC, which sets up a user's DataRefresher library in the VM environment. Before you can run DVRELNRL, you must edit it. For information on editing DVREXLIB, see “Editing VM EXECs for your environment” on page 166.
Calling the DVREXLIB EXEC

Use the following steps to call the DVREXLIB EXEC:

1. Log on to the administrator's user ID.
2. Access the DataRefresher production disk.
3. Pass the following parameters to the EXEC, or be aware of them when prompted by the EXEC (if you are enrolling yourself as administrator, you are not prompted for all the parameters):
   - **parm1** The user ID of the user being enrolled. If you are the administrator, this will be the same as the user ID on which you are calling this EXEC. This user ID is used for the filename of the profile MACLIB.
   - **parm2** The filemode of the administrator's disk on which the profile and DVRJEDIE to be copied exists.
   - **parm3** Either a Y or an N, depending on whether the user you enroll was a previous DataRefresher dialogs user. Y indicates a user was a DXT dialogs user.
   - **parm4** The JCL and edited objects filename. The DataRefresher default is DVRJEDIE MACLIB.

Use the following command and syntax to call the EXEC passing the parameters:

```
EXEC DVREXLIB parm1 parm2 parm3 parm4
```

or use the following to call the EXEC and be prompted for the parameters:

```
EXEC DVREXLIB
```

How the EXEC runs

If you are enrolling:

- The DataRefresher administrator, enter the administrator's user ID, and press ENTER. If your do not want to enter other users, press ENTER again in response to the prompt.
- Users other than the administrator, the EXEC prompts you for their user IDs. For each user you want to enroll, type the user ID and press ENTER. When you have enrolled all users, press ENTER in response to the prompt.

When the EXEC is complete, the DataRefresher dialogs libraries should have been built and sent to each user being enrolled.

For information on using DataRefresher dialogs, see the DataRefresher MVS and VM User's Guide.

DataRefresher dialogs models

DataRefresher models provide you with examples of how to complete a number of DataRefresher tasks. You can find the models for use on VM in a macro library called DVRJEDIE MACLIB. If you would like to see a model, you can:

- Call the model up during a DataRefresher dialogs session
- Browse the model outside the dialogs, as you would any other data set to which you have access
• Print the model using the procedure you use to print a data set
• Browse and edit the model while using DataRefresher online commands

The models contain instructions to change the model to suit your needs.

Data description models

Table 32 describes the data description models provided with DataRefresher. These models are located in the macro library DVRJEDIE MACLIB.

<table>
<thead>
<tr>
<th>MODEL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVREDREF</td>
<td>Creates a DXTFILE description</td>
</tr>
<tr>
<td>DVREDREP</td>
<td>Creates a DXTPSB description</td>
</tr>
<tr>
<td>DVREDCRD</td>
<td>Creates a DXT DATATYPE type description</td>
</tr>
<tr>
<td>DVREDRVF</td>
<td>Creates a DXTVIEW of a DXTFILE description</td>
</tr>
<tr>
<td>DVREDRVP</td>
<td>Creates a DXTVIEW of a DXTPCB description</td>
</tr>
</tbody>
</table>

Extract request models

Table 33 describes the extract request models provided with DataRefresher. These models are located in the macro library DVRJEDIE MACLIB.

<table>
<thead>
<tr>
<th>MODEL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVREDRXV</td>
<td>Submits REM extract requests in VM using the SUBMIT(REM) command</td>
</tr>
<tr>
<td>DVREDEXT</td>
<td>Submits DEM extract requests using the SUBMIT(UIM) command</td>
</tr>
</tbody>
</table>

JCL/JCS models

Table 34 describes the JCL and JCS models provided with DataRefresher. These models are located in the macro library DVRJEDIE MACLIB.

<table>
<thead>
<tr>
<th>MODEL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVREDREM</td>
<td>Starts the REM when extracting from a DB2 source</td>
</tr>
<tr>
<td>DVREDREV</td>
<td>Starts the REM in VM</td>
</tr>
<tr>
<td>DVREDDAP</td>
<td>Starts the DAP</td>
</tr>
<tr>
<td>DVREDDXT</td>
<td>Starts the DXTINPUT procedure</td>
</tr>
<tr>
<td>DVREDDJC</td>
<td>Loads extracted data into DB2</td>
</tr>
<tr>
<td>DVREDDBC</td>
<td>Creates and loads into DB2</td>
</tr>
<tr>
<td>DVREDSJC</td>
<td>Loads extracted data into SQL/DS</td>
</tr>
<tr>
<td>DVREDRUV</td>
<td>Executes the DRU in VM</td>
</tr>
<tr>
<td>DVREDEJV</td>
<td>Executes DataRefresher's MIT utility in VM</td>
</tr>
</tbody>
</table>
SAP Skeletons

Table 35 describes the SAP skeleton models provided with DataRefresher. These models are located in the macro library DVRJEDIE MACLIB.

<table>
<thead>
<tr>
<th>NAME</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVRXKDJA</td>
<td>SAP dialog (CREATE DXTPSB)</td>
</tr>
<tr>
<td>DVRXKJFA</td>
<td>SAP dialog (CREATE DXTFILE)</td>
</tr>
</tbody>
</table>

Maintaining library space for your users

As administrator, you must maintain library space allocations for the DataRefresher users at your site. If there is insufficient space available, users may receive a warning message in the middle of their DataRefresher dialogs session. If you operate under VM, maintain library space allocations using the ISPF library COMPRESS utility to reorganize a MACLIB.

Administering the End User Dialogs

Following are task overviews for building a nickname table, updating an end user table, and requesting source descriptions (updating a master index table). For full DataRefresher dialogs operating procedures, see the DataRefresher MVS and VM User's Guide.

Building the nickname table

1. Log on to Administrative Dialogs
2. From the main menu, select option 5 to display the END USER DIALOGS ADMINISTRATION panel
3. Select option 1 to display the ESTABLISH NICKNAMES FOR SYSTEMS panel
4. Specify information for nicknames

Updating the end user table

1. Log on to Administrative Dialogs
2. From the main menu, select option 5 to display the END USER DIALOGS ADMINISTRATION panel
3. Select option 2 to display the UPDATE THE END USER TABLE panel
4. Fill in appropriate source node ID, source user ID, and DataRefresher dialogs user ID

Requesting source descriptions

1. Log on to Administrative Dialogs
2. From the main menu, select option 5 to display the END USER DIALOGS ADMINISTRATION panel
3. Select option 3 to display the REQUEST SOURCE TABLE DESCRIPTION(S) panel
4. Fill in fields on REQUEST SOURCE TABLE DESCRIPTION(S) panel
5. Specify tables or DataRefresher views you want
6. Review JCL and Load JCS, if necessary
7. Review job stream if necessary
8. Submit job to the REM
Part 4. Appendixes, terms and abbreviations
Appendix A. MVS installation worksheets

The worksheets in this section will help you prepare for installing DataRefresher with the DataRefresher Installation Feature.

DataRefresher base product installation worksheet (MVS)

<table>
<thead>
<tr>
<th>Information requested</th>
<th>Default value supplied</th>
<th>New value</th>
</tr>
</thead>
<tbody>
<tr>
<td>High level qualifier for DataRefresher data sets *</td>
<td>DVR110</td>
<td></td>
</tr>
<tr>
<td>High level qualifier for SMP CSI, log data sets *</td>
<td>SMPPRE</td>
<td></td>
</tr>
<tr>
<td>High level qualifier for DataRefresher distribution data sets *</td>
<td>DVR110</td>
<td></td>
</tr>
<tr>
<td>Distribution medium for DataRefresher</td>
<td>3480</td>
<td></td>
</tr>
<tr>
<td>Unit where DataRefresher data sets are to reside</td>
<td>SYSDA</td>
<td></td>
</tr>
<tr>
<td>Volume on which DataRefresher data sets are to reside</td>
<td>VOLID</td>
<td></td>
</tr>
<tr>
<td>Name of the TSO Terminal monitor program</td>
<td>IKJEFT01</td>
<td></td>
</tr>
<tr>
<td>Accounting and user information for the batch jobs the installation EXEC will submit</td>
<td>(ACCT#), 'USERID'</td>
<td></td>
</tr>
<tr>
<td>MVS job class for jobs using 15 minutes or less of CPU time</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>MVS job class for jobs with no limit on the CPU time</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>MVS MSGCLASS parameter for the batch jobs the installation EXEC will submit</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>MVS MSGLEVEL parameter for the batch jobs the installation EXEC will submit</td>
<td>1,1</td>
<td></td>
</tr>
<tr>
<td>Unit where SMP/E data sets will reside</td>
<td>SYSDA</td>
<td></td>
</tr>
<tr>
<td>Volume where SMP/E data sets will reside</td>
<td>VOLID</td>
<td></td>
</tr>
<tr>
<td>Unit on which DataRefresher distribution data sets will reside</td>
<td>SYSDA</td>
<td></td>
</tr>
<tr>
<td>Volume on which DataRefresher distribution data sets will reside</td>
<td>VOLID</td>
<td></td>
</tr>
<tr>
<td>High level qualifier of ISPF program product</td>
<td>ISR</td>
<td></td>
</tr>
<tr>
<td>High level qualifier of PDF program product</td>
<td>ISR</td>
<td></td>
</tr>
</tbody>
</table>
Table 36 (Page 1 of 2). Information required to install the DataRefresher base product on MVS. Items marked with an asterisk (*) require new installation information.

<table>
<thead>
<tr>
<th>Information requested</th>
<th>Default value supplied</th>
<th>New value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume for temporary dataset required during SMP/E processing</td>
<td>VOLID</td>
<td>N</td>
</tr>
<tr>
<td>Low level qualifier of CLIST dataset allocated to SYSPROC *</td>
<td>DVRCLIBE</td>
<td></td>
</tr>
<tr>
<td>Low level qualifier of REXX dataset allocated to SYSPROC *</td>
<td>REXX.EXEC</td>
<td></td>
</tr>
<tr>
<td>Fully qualified name of output dataset for installation verification of the DRU *</td>
<td>DVR.DRUIVP</td>
<td></td>
</tr>
<tr>
<td>Migration choice - Y if migrating from DXT V2R4 *</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Unit of CLIST dataset allocated to SYSPROC</td>
<td>SYSDA</td>
<td></td>
</tr>
<tr>
<td>Volume ID of CLIST dataset allocated to SYSPROC</td>
<td>VOLID</td>
<td></td>
</tr>
<tr>
<td>Unit of REXX dataset allocated to SYSPROC</td>
<td>SYSDA</td>
<td></td>
</tr>
<tr>
<td>Volume of REXX dataset allocated to SYSPROC</td>
<td>VOLID</td>
<td></td>
</tr>
<tr>
<td>Unit for the output dataset for the DRU IVP</td>
<td>SYSDA</td>
<td></td>
</tr>
<tr>
<td>Volume for the output dataset for the DRU IVP</td>
<td>VOLID</td>
<td></td>
</tr>
<tr>
<td>Node ID of machine where DataRefresher is being installed</td>
<td>NODEID</td>
<td></td>
</tr>
<tr>
<td>Low level qualifiers for ISPF and PDF program product libraries</td>
<td>ISPLOAD</td>
<td>ISPLOAD</td>
</tr>
<tr>
<td></td>
<td>ISRLIB</td>
<td>ISRLIB</td>
</tr>
<tr>
<td></td>
<td>ISPsMLIB</td>
<td>ISPsMLIB</td>
</tr>
<tr>
<td></td>
<td>ISPPLIB</td>
<td>ISPPLIB</td>
</tr>
<tr>
<td></td>
<td>ISRPPLIB</td>
<td>ISRPPLIB</td>
</tr>
<tr>
<td></td>
<td>ISPSLIB</td>
<td>ISPSLIB</td>
</tr>
<tr>
<td></td>
<td>ISRSLIB</td>
<td>ISRSLIB</td>
</tr>
<tr>
<td></td>
<td>ISPTLIBM</td>
<td>ISPTLIBM</td>
</tr>
<tr>
<td></td>
<td>ISRITLIB</td>
<td>ISRITLIB</td>
</tr>
<tr>
<td></td>
<td>JCLBASE1</td>
<td>JCLBASE1</td>
</tr>
<tr>
<td></td>
<td>JCLUPPE1</td>
<td>JCLUPPE1</td>
</tr>
<tr>
<td></td>
<td>JCLKANJ1</td>
<td>JCLKANJ1</td>
</tr>
</tbody>
</table>
### General Data Extract feature installation worksheet (MVS)

**Table 37. Information required to install the General Data Extract feature on MVS. Items marked with an asterisk (*) require new installation information.**

<table>
<thead>
<tr>
<th>Information requested</th>
<th>Default value supplied</th>
<th>New value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low level qualifier of dataset for sample data *</td>
<td>DVR.SAMPDATA</td>
<td></td>
</tr>
<tr>
<td>Low level qualifier for DataRefresher EXTLIB *</td>
<td>EXTLIB</td>
<td></td>
</tr>
<tr>
<td>Low level qualifier for DataRefresher FDTLIB sample data *</td>
<td>FDTLIB</td>
<td></td>
</tr>
<tr>
<td>Will the installation require exit routines written in PL/I? *</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Will the installation require exit routines written in COBOL? *</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>PL/I libraries for compiling and linking</td>
<td>SYS1.PLIBASE</td>
<td>SYS1.SIBMBASE</td>
</tr>
<tr>
<td>PL/I run time libraries</td>
<td>SYS1.PLILINK</td>
<td>SYS1.SIBMSE</td>
</tr>
<tr>
<td>High level qualifier of IMS/VS program product libraries</td>
<td>IMSVS</td>
<td></td>
</tr>
<tr>
<td>High level qualifier of Data Dictionary program product libraries</td>
<td>IMSVS.DD</td>
<td></td>
</tr>
<tr>
<td>COBOL library</td>
<td>SYS1.COBLIB</td>
<td></td>
</tr>
<tr>
<td>Block size of DBD and PSB object libraries</td>
<td>3120</td>
<td></td>
</tr>
</tbody>
</table>
Relational Data Extract feature Installation Worksheet (MVS)

Table 38. Information required to install the DataRefresher Relational Data Extract feature on MVS. Items marked with an asterisk (*) require new installation information.

<table>
<thead>
<tr>
<th>Information requested</th>
<th>Default value supplied</th>
<th>New value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage group for DB2 sample data *</td>
<td>DVR1STBG</td>
<td></td>
</tr>
<tr>
<td>Name of DB2 database for sample data *</td>
<td>DVR1STBB</td>
<td></td>
</tr>
<tr>
<td>Name of DB2 tablespace for sample data *</td>
<td>DVR1STBT</td>
<td></td>
</tr>
<tr>
<td>Names for three tables for sample data *</td>
<td>DVR,DB2ORG</td>
<td>DVR,DB2ORG</td>
</tr>
<tr>
<td></td>
<td>DVR,DB2STAFF</td>
<td>DVR,DB2PROJ</td>
</tr>
<tr>
<td>REM plan *</td>
<td>DVRPLAN</td>
<td></td>
</tr>
<tr>
<td>DB2 run time library</td>
<td>DSN230.DSNLOAD</td>
<td></td>
</tr>
<tr>
<td>DB2 samples run time library</td>
<td>DSN230.RUNLIB.LOAD</td>
<td></td>
</tr>
<tr>
<td>DSN subsystem</td>
<td>DSN</td>
<td></td>
</tr>
<tr>
<td>DB2 sample dynamic SQL program</td>
<td>DSNTIAD</td>
<td></td>
</tr>
<tr>
<td>DB2 Plan</td>
<td>DSNTIAD</td>
<td></td>
</tr>
<tr>
<td>Volume for creating DB2 storage group</td>
<td>VOLID</td>
<td></td>
</tr>
<tr>
<td>Catalog for creating DB2 storage group</td>
<td>DSNV</td>
<td></td>
</tr>
</tbody>
</table>
# Uppercase and Kanji Features installation worksheet (MVS)

## Uppercase Feature

Table 39. Information required to install the DataRefresher Uppercase Feature on MVS. Items marked with an asterisk (*) require new installation information.

<table>
<thead>
<tr>
<th>Information requested</th>
<th>Default value supplied</th>
<th>New value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of CLIST library allocated to SYSPROC *</td>
<td>DVRCLIBU</td>
<td></td>
</tr>
</tbody>
</table>

## Kanji Feature

Table 40. Information required to install the DataRefresher Kanji Feature on MVS. Items marked with an asterisk (*) require new installation information.

<table>
<thead>
<tr>
<th>Information requested</th>
<th>Default value supplied</th>
<th>New value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of CLIST library allocated to SYSPROC *</td>
<td>DVRCLIBK</td>
<td></td>
</tr>
</tbody>
</table>
Appendix B. VM installation worksheet

The following information may be needed to install DataRefresher on your VM operating system:

- The link/access information is one of the following depending on whether the disk storage method chosen is minidisk or shared file minidisk:
  - LINK DXTPROD 200 200 - ACC 200 P for non-shared file access
  - ACC .DXTPROD P for shared file access

- The SQL/DS version/release should be indicated as 2 characters. For example, Version 1 Release 0 would be indicated as 10.

- The SQL/DS connect password for DataRefresher can be any password that is acceptable to SQL/DS. The DataRefresher user ID is created if the Relational Data Extract feature is installed and when DVREXRPR EXEC is run.

Fill in the blanks in the VM Installation Worksheet with the information that applies to your installation. You will be prompted for these values during the install process.
<table>
<thead>
<tr>
<th>VM Installation Worksheet</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DataRefresher base product installation</strong></td>
</tr>
<tr>
<td>DataRefresher distribution disk link/access</td>
</tr>
<tr>
<td>DataRefresher production disk link/access</td>
</tr>
<tr>
<td>ISPF disk link/access</td>
</tr>
<tr>
<td><strong>DXT master index table migration</strong></td>
</tr>
<tr>
<td><strong>DataRefresher Relational Data Extract feature installation</strong></td>
</tr>
<tr>
<td>SQL/DS connect password for previous userid</td>
</tr>
<tr>
<td>SQL/DS connect password for user ID DataRefresher</td>
</tr>
<tr>
<td>SQL/DS version/release</td>
</tr>
<tr>
<td>SQL/DS database name</td>
</tr>
<tr>
<td>SQL/DS disk link/access</td>
</tr>
<tr>
<td><strong>DataRefresher UCF installation</strong></td>
</tr>
<tr>
<td><strong>DataRefresher Kanji Feature installation</strong></td>
</tr>
</tbody>
</table>
Appendix C. Task overviews for administrators

The following tables provide you with task overviews for administering the:

- General Data Extract feature
- Relational Data Extract feature
- DataRefresher online commands
- DataRefresher dialogs

Administering the General Data Extract feature

Following are task overviews for administering the General Data Extract feature in a dialogs and non-dialogs environment.

Task overview—dialogs environment

1. Create an FDTLIB and an EXTLIB
2. Set up DEM operation
3. Implement a data-protection scheme
4. Set up DataRefresher OS/2 operation (optional)
5. Enroll the administrator
6. Perform End User Dialogs administration
7. Enroll users
8. Create data descriptions
9. Administer the DAP (optional)
10. Administer the SAP (optional)
11. Edit DataRefresher models to suit your site's requirements
12. Write exit routines (optional)

Task overview—non-dialogs environment

1. Create an FDTLIB and an EXTLIB
2. Set up DEM operation
3. Implement a data-protection scheme
4. Set up DataRefresher OS/2 operation (optional)
5. Create data descriptions
6. Administer the DAP (optional)
7. Administer the SAP (optional)
8. Edit DataRefresher models to suit your site's requirements
9. Write exit routines (optional)
10. Administer the DataRefresher online commands (optional)
Administering the Relational Data Extract feature

Following are task overviews for administering the Relational Data Extract feature in a dialogs and non-dialogs environment.

Task overview—dialogs environment

1. Implement a data-protection scheme
2. Enroll the administrator
3. Perform End User Dialogs administration
4. Enroll users
5. Edit DataRefresher models to suit your site's requirements

Task overview—non-dialogs environment

1. Implement a data-protection scheme
2. Administer the DataRefresher online commands (optional)
3. Edit DataRefresher models to suit your site's requirements

Administering the DataRefresher online commands

Following is the task overview for administering the DataRefresher online commands.

1. Ensure all necessary DataRefresher features are installed
2. Set up environment to point to the DataRefresher online commands library
3. Create/modify default parameters
4. Create the required libraries
5. Tailor DataRefresher online commands to suit your site's requirements

Administering the End User Dialogs

Following are task overviews for building a nickname table, updating an end user table, and requesting source descriptions (updating a master index table).

Task overview—build nickname table

1. Log on to Administrative Dialogs
2. From the main menu, select option 5 to display the END USER DIALO GS ADMINISTRATION panel
3. Select option 1 to display the ESTABLISH NICKNAMES FOR SYSTEMS panel
4. Specify information for nicknames
Task overview—update end user table

1. Log on to Administrative Dialogs
2. From the main menu, select option 5 to display the END USER DIALOGS ADMINISTRATION panel
3. Select option 2 to display the UPDATE THE END USER TABLE panel
4. Fill in appropriate source node ID, source user ID, and DataRefresher dialogs user ID.

Task overview—request source descriptions

1. Log on to Administrative Dialogs
2. From the main menu, select option 5 to display the END USER DIALOGS ADMINISTRATION panel
3. Select option 3 to display the REQUEST SOURCE TABLE DESCRIPTION(S) panel
4. Fill in fields on REQUEST SOURCE TABLE DESCRIPTION(S) panel
5. Specify tables or DXTVIEW views you want
6. Review JCL and Load JCS, if necessary
7. Review job stream if necessary
8. Submit job to UIM or REM
Appendix D. DataRefresher performance considerations

This chapter discusses some of the factors affecting the performance of DataRefresher.

- “Surveying DataRefresher performance” on page 189 provides information to help you assess both system and DataRefresher performance.
- “Specifying data selection criteria in an extract request” on page 190 describes how data selection criteria in an extract request can affect extract performance.
- “Joining data from multiple sources” on page 196 provides information on how to join data sources and sustain DataRefresher performance.
- “Methods to speed extract processing” on page 200 provides information on how to improve extract processing by using keyed data sets and batching extract requests.
- “Tips for better DataRefresher performance” on page 202 provides a list of general rules to follow to enhance and sustain DataRefresher performance.

Surveying DataRefresher performance

DataRefresher's performance is closely tied to the performance of your host system.

System performance

If you suspect a performance problem, first check the following areas of your system:

- **Central Processing Unit (CPU):** CPU usage affects all systems and can cause poor performance and response times.
- **Paging:** Excessive paging downgrades performance. Sometimes a site uses storage fencing to favor certain systems (for example, CICS or IMS/DC), but this storage fencing leaves less time for batch work. Because most DataRefresher processing is run in batch, DataRefresher can seem slow.
- **Disk Input/Output (I/O):** Long response time for I/O operations affects all applications that use direct access storage devices (DASDs).

If you think that DataRefresher performance could be improved, first consult your system programmer to check these areas of your system. With an understanding of these three areas of your system, your system programmer can ask:

- How heavily is the CPU used?
- How many I/Os are taking place per file?
- Are these figures reasonable?
DataRefresher performance

If your system is working well, the following considerations may help you locate sources of downgraded DataRefresher performance:

- What part of DataRefresher has downgraded performance?
  - DataRefresher dialogs
  - UIM when storing extract requests
  - DEM when extracting from non-relational databases
  - REM when extracting from DB2 databases

- If you consider the DEM extracts to have affected performance, are all DEM runs too long or just one or two? (If all DEM runs are too long, there is a system problem.)

- Is your DataRefresher problem new? Was everything working properly recently? If so, what has changed?

- Can you monitor the part of DataRefresher with downgraded performance? Other subsystems are used by DataRefresher. Each subsystem should have a performance tool that can be used to analyze a DataRefresher's use of that subsystem.

For example, the DB2 database has the DB2PM tool, which creates a report by user regarding the resources used for REM extracts. Also, a DEM run that uses IMS can be changed to call and use the IMS monitor to report how the DEM performed.

- Is your extract request efficiently coded?

Specify data selection criteria in an extract request

Selection criteria in an extract request can affect extract performance. For example, ineffective criteria can cause DataRefresher to search the entire data set for a small amount of extract data.

Specify key fields in the WHERE clause

The WHERE clause, part of the SELECT statement in the extract request, specifies selection criteria and can limit the data which qualifies to be extracted.

For example, suppose you want to extract from the following data set all records that contains a value from 105500 to 105599 in fields KEY and FIELD.

<table>
<thead>
<tr>
<th>KEY</th>
<th>FIELD</th>
<th>FIELD</th>
</tr>
</thead>
<tbody>
<tr>
<td>001256</td>
<td>207899</td>
<td>.</td>
</tr>
<tr>
<td>003658</td>
<td>001256</td>
<td>.</td>
</tr>
<tr>
<td>007899</td>
<td>105512</td>
<td>.</td>
</tr>
<tr>
<td>037890</td>
<td>089811</td>
<td>.</td>
</tr>
<tr>
<td>068963</td>
<td>145822</td>
<td>.</td>
</tr>
<tr>
<td>089811</td>
<td>102333</td>
<td>.</td>
</tr>
<tr>
<td>102333</td>
<td>127899</td>
<td>.</td>
</tr>
<tr>
<td>105512</td>
<td>003658</td>
<td>.</td>
</tr>
<tr>
<td>105587</td>
<td>089811</td>
<td>.</td>
</tr>
<tr>
<td>127899</td>
<td>007890</td>
<td>.</td>
</tr>
<tr>
<td>145822</td>
<td>105587</td>
<td>.</td>
</tr>
<tr>
<td>207899</td>
<td>068811</td>
<td>.</td>
</tr>
</tbody>
</table>

DataRefresher uses the criteria you specify on the WHERE clause to formulate its calls to the access method.
For improved performance, use a key field wherever possible. For example, in this keyed data set, the same data is coded in two columns:

- As ordered values in the KEY field
- As random values in the FIELD A field

Thus, you can write a WHERE clause specifying the KEY field as follows:

```
WHERE KEY BETWEEN 105500 AND 105599
```

In this example, the WHERE clause refers to the KEY field. DataRefresher knows it is a key field, and reads the required data in two calls to find the valid records (105512 and 105587).

However, if the WHERE clause referred to FIELD A, or DataRefresher does not recognize the field you specify as a key sequence field, DataRefresher has to read every record in the data set and compare the contents with the criteria.

**Specifying key fields in CREATE commands**

Always use a keyed field for selective extracts. You define key fields to DataRefresher in the CREATE DXTFILE and CREATE DXTPSB commands using the following keywords:

- `SEQUENCE=ASC` or `DESC`
- `UNIQUE=Y` or `N`
- `SEQFLD=R` or `V`

These keywords are optional, but you should code them for better performance for selective extracts or for extracts involving more than one data set. If you specify that a field is sequenced, DataRefresher relies on that information. However, if you do not specify, for example, that a particular field is the primary key (`SEQFLD=R`), DataRefresher assumes that the CREATE DXTFILE command refers to a non-sequenced data set.

**SEQUENCE=ASC or DES**

This keyword tells DataRefresher whether the next value read will be higher or lower. It has meaning only with SAM data sets or VSAM ESDS data sets.

**UNIQUE=Y or N**

This keyword tells DataRefresher that the key is unique. When DataRefresher is searching for a range of values in a data set and the last wanted value is returned by the access method, DataRefresher must still ask for the next record to check for duplicates. If you specify this keyword, the extra call is eliminated. If the data set is joined to another and used as an inner data set (See “Joining data using inner and outer loops” on page 198) the number of reads could be reduced by half.

**SEQFLD=R or V**

This keyword is useful when a range of keys is specified in the WHERE clause. It allows DataRefresher to start a search at a low key value and stop searching when a value has been reached that is greater than the required key value(s). You must specify at least the SEQFLD keyword to get performance benefits when extracting from key fields. If you do not specify SEQFLD, DataRefresher treats all requests as if they were for flat files and assumes that the data is in random order.
Thus, if you do not specify a key sequence field, even a request from one record using `FIELDA=xxxx` on the root primary key for a VSAM KSDS would result in a read of the entire data set.

**Sequential data sets:** If a data set is a SAM or VSAM ESDS sequential data set and is stored in physical sequence, specifying `SEQFLD=R` increases efficiency if less than the entire data set is needed. DataRefresher must read sequentially from the start of the sequential data set. When the key of the returned record is higher than the value specified in the `WHERE` clause (where `FIELDA=xxxx`), DataRefresher stops reading the data set.

This can result in a considerable reduction of I/Os, especially if two data sets are joined and the inner data set is in sequential format. See “Joining data using inner and outer loops” on page 198.

**Valid key fields for use with the SEQFLD keyword:** Specify the SEQFLD keyword in the `CREATE DXTFILE` and `CREATE DXTPSB` commands to get the performance benefits of using key sequence fields. Using a primary key lets DataRefresher go directly to the data that is required for your extract.

Specifying a key field with SEQFLD implies that the field is the:

- Primary or alternate key for a VSAM KSDS.
- Alternate key of a VSAM ESDS.
- A field of a SAM data set or VSAM ESDS with data arranged in the order of that field.
- Primary key of an internal segment with data arranged in the order of that field.
- Primary key of an IMS database using one of the following access methods:
  - HSAM or SHSAM
  - HISAM or SHISAM
  - HIDAM
  - MSDB

Specifying a primary key of an IMS HDAM or IMS DEDB database is valid, but it may not be useful. See "Key fields that are not especially useful with the SEQFLD keyword" on page 193 for more information.

- Primary key of a dependent IMS segment that is designated to IMS as a key field.
- Key used in an IMS secondary index.

**IMS dependent segments:** Dependent segments of all IMS databases can be read successfully using GET NEXT processing. Thus, within a particular database record, key fields are always useful.

If you specify a join using two views of the same PCB so that part of the hierarchical path is the same for both views, DataRefresher performs a join at the lowest common level.

For example, consider the following database or data set hierarchy and views, VIEW1 and VIEW2.
If you specify the FROM keyword (in the WHERE clause) in an extract request as, FROM VIEW1, VIEW2, DataRefresher adds an implied condition that the WHERE VIEW1 occurrence of B is the same as the VIEW2 occurrence of B when C, D, or E are the target segments.

This is also true for a SAM or VSAM data set with internal segments if more than one leg exists in the data set structure.

**Invalid key fields for the SEQFLD keyword:** Only specify SEQFLD for a valid sequence field. If you specify to DataRefresher that a field is a sequence field when it is not, or if you specify to DataRefresher that the field values are unique when there are duplicates, all extract requests using this information are invalid.

Again, if you specify a range of values for a field that is not sequenced, DataRefresher stops searching as soon as it finds a value higher than that specified in the range. Thus, many records may be left unread.

**Key fields that are not especially useful with the SEQFLD keyword**
Using certain key fields may have little effect upon DataRefresher performance at your site.

**IMS keys:** Primary keys are not always useful with the following IMS databases:

- IMS HDAM
- IMS DEDB

These types of IMS databases use a randomizer to store the database records. Because the database records are stored in random order, all attempts using GET NEXT sequential processing will result in the records being returned in random sequence.

For example, if you specify a WHERE clause for data from an IMS HDAM or IMS DEDB database as, WHERE FIELD1 = 12345678, DataRefresher can use IMS services to get that particular record (12345678).
DataRefresher will also use IMS services for locating a record if you specify a WHERE clause listing a number of discrete values, such as, IN (xxxx, yyyy, zzzz).

However, if you specify a WHERE clause that uses a greater than (> ) condition or a range of values, DataRefresher must read the entire IMS database to check for all possibilities even if only a few values qualify. This would result in extra I/Os for the small amount of data.

For a discussion of ways to deal with this and other selection problems, see “Methods to speed extract processing” on page 200.

**IMS secondary indexes:** Use IMS primary keys, in preference to keys using secondary indexes, in cases where the database is not HDAM.

Consider the secondary index Secondary Index A in Figure 42 on page 194. This figure shows a database keyed on the personnel number, with a secondary index created to give access if only the name is known.

![IMS secondary index diagram]

Figure 42. IMS secondary index

If a secondary index is used, reading each record in the database will usually produce an I/O, because the next record required will be in a different block. Whereas when using a primary key, the next record required will, in most cases, be adjacent to the one before. Records are read in blocks, and therefore the required record will already be in storage.

So, reading the entire database with the primary key rather than a secondary index will be more efficient by a factor equal to the blocking factor.

SEQFLD=V is used only where an IMS secondary index is used to read the data, and the source field is not in the same segment as the target data (as shown in the previous figure) for Secondary Index B.

Figure 42 on page 194 shows the Secondary Index B as a secondary index which will return a root segment with a person’s name when only the skill is known. This key is derived from a dependent segment.

Normally, when the SELECT statement specifies fields to be returned by DataRefresher, all the fields will be present in the data segment. However, with a
secondary index, where the target is not the source, the segment does not contain the key.

You can, however, define the key field as part of the DXTPCB segment and indicate that it is not a true field by omitting a START keyword. You can direct DataRefresher to return the key field as part of the SELECT statement by naming this field. This is only possible if the IMS PCB was generated with the PROCSEQ option, indicating use of a secondary index. Consult your IMS database administrator regarding whether the PCB uses a secondary index.

When processing an IMS database through the use of a secondary index, the index key value is returned in the IMS PCB key feedback area. Specifying SEQFLD=V directs DataRefresher to find this field in the IMS PCB key feedback area, not in the segment. In the previous figure, the secondary index field is the skill.

**VSAM:** Reading an entire file is much more efficient with a primary key. Specifying SEQFLD=V does not apply to VSAM.

**Numeric keys:** DataRefresher can use the access method services of IMS or VSAM if an equate (‘=’ or ‘IN’) operator is used to get that particular record.

For example, if you specify a WHERE clause for data from an IMS database as, WHERE FIELD1 = 12345678, DataRefresher can use IMS services to get that particular record (12345678). DataRefresher will also use IMS or VSAM services for locating a record if you specify a WHERE clause listing a number of discrete values, such as, IN (xxxx, yyyy, zzzz).

However, if you specify a WHERE clause that uses any other operator or a range of values, DataRefresher must read the entire data source to check for all possibilities even if only a few values qualify. This would result in extra I/Os for the small amount of data.

**Specifying appropriate fields in the SELECT statement**

For improved performance of selective extracts, you should specify appropriate fields in the SELECT statement.

**IMS databases:** If you extract from an IMS database, always use a primary key. If you specify only a dependent field, DataRefresher must test all of the dependents.

For logical database descriptions (DBDs), ensure that IMS databases are linked properly to support your request. Selection of data from a logically linked database is more efficient if the concatenated key of the logical parent is stored in the logical child.

See your IMS database administrator for more information.

**DB2 databases:** If you extract from a DB2 database, always extract from key fields and ensure that there is an index on the column used for the extract. Ideally this index should be a clustering index.

If a column has a clustering index, then the data is retrieved by DB2 in order of that field, and additionally, the data is in the correct order in the block read. Thus,
clustering indexes represent a significant performance improvement over the use of an ordinary index.

See your DB2 database administrator for more information.

Joining data from multiple sources

To join data from multiple data sources, specify a DXTVIEW for each source, and specify selection criteria to tell DataRefresher how to join the data.

For example, if you want to extract from fields A, B, C, D, and E (where A and D are key fields) using views X and Y you could write a SELECT statement as follows:

```
SELECT   A, B, C, D, E
FROM      X, Y
WHERE     A = D
```

In this example, key fields are used to join the two views X and Y.

The way in which data sets are joined can affect DataRefresher performance when extracting data.

When joining data sets from two or more sources (up to 16 sources can be joined), DataRefresher selects the most efficient way to access the data to minimize processing and record retrievals. However, DataRefresher relies on the proper keywords being specified in the CREATE DXTFILE and DXTPSB commands, and on the most appropriate use of the join parameters.

To join your data, DataRefresher will either use:

- Parallel access processing
- Inner loop processing

Joining sequential data using parallel access

When you join two or more of the following sequentially accessed data sets or databases:

- Physical sequential data set
- VSAM ESDS
- GDI record
- DL/I HSAM or SHSAM database
- DL/I Fast Path MSDB database

DataRefresher will join these data sources using parallel access if they:

- Are joined using WHERE clause equal (=) operators on their major key/sequence fields (or prefixes if one of the character or graphic data types)
- Are being joined in the same ascending or descending sequence

When you join two or more data sources, only one of the data sets can have a non-unique key. When you join data sets with non-unique keys, you must specify UNIQUE=N for those data sets, in order to avoid losing records. The default for the UNIQUE keyword is Y.
For example, suppose you want to join and extract data from two physical sequential data sets, F1 and F2, using their respective keyed fields, F1SORT and F2SORT, as shown in Figure 43 on page 197.

```
DATA SET F1 containing
   F1SORT Field
AAABCD...
AAABXY...
AAABAAA...
ABCKXYZ...
ABCKZZZ...
ABFCCCC...
ABFDDD...
ABFEEEE...
BBBXYZ...
BBDXYZ...
BXXXYZ...
CBCKXYZ...
CCCCXYZ...
CCHXYZ...
CCHZZZZ...
DAAAAA...
```

```
DATA SET F2 containing
   F2SORT Field
AAAB...
AAAF...
AAAG...
ABF...
ABG...
ABH...
AZZ...
CCC...
CCD...
CCE...
CCG...
CHH...
CGG...
CHA...
```

Figure 43. Data sets F1 and F2

You could write DXTFILE descriptions defining data sets F1 and F2 as follows:

```
CREATE DXTFILE NAME=F1,
   ACCESS=PS
   FIELD NAME=F1SORT, BYTES=6, START=1, SEQFLD=R,
   FIELD NAME=F1SORTPF, BYTES=3, START=1, SEQFLD=R, UNIQUE=N, 1
   FIELD NAME=F1DATA1, ...
   ...

1 The F1SORTPF field is the three-character prefix of the key field F1SORT.
```

```
CREATE DXTFILE NAME=F2,
   ACCESS=PS
   FIELD NAME=F2SORT, BYTES=3, START=1, SEQFLD=R,
   FIELD NAME=F2DATA1, ...
   ...
```

Based on these DXTFILE descriptions, you could define the DataRefresher views for F1 and F2 as follows:

```
CREATE DXTVIEW NAME=V1,
   DXTFILE=F1,
   FIELDS=*
```

```
CREATE DXTVIEW NAME=V2,
   DXTFILE=F2,
   FIELDS=*
```

Given the data descriptions defined for data sets F1 and F2, you could write the following extract request:
WHERE F2SORT=F1SORTPF specifies that the F2 key field be equal to a prefix of the F1 key field. If the key fields are the same length, they would have to be related in the WHERE clause to provide parallel access.

Figure 44 on page 198 shows how parallel access processing occurs when joining data sets F1 and F2. The numbers in the figures indicate the order of access. The arrows indicate the reads after which an extract from the joined last records from each data set occurs.

<table>
<thead>
<tr>
<th>DATA SET F1</th>
<th>DATA SET F2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ABABCD...</td>
<td>2 AAB...</td>
</tr>
<tr>
<td>3 AABXY...</td>
<td>6 AAF...</td>
</tr>
<tr>
<td>4 AABAAAA...</td>
<td>7 AAG...</td>
</tr>
<tr>
<td>5 ABCXYZ...</td>
<td>8 ABF...</td>
</tr>
<tr>
<td>9 ABXXZ...</td>
<td>14 ABB...</td>
</tr>
<tr>
<td>10 ABBCCCC...</td>
<td>15 ABH...</td>
</tr>
<tr>
<td>11 ABBFFFF...</td>
<td>16 AZZ...</td>
</tr>
<tr>
<td>12 ABBEEEE...</td>
<td>17 CCC...</td>
</tr>
<tr>
<td>13 BBBXYZ...</td>
<td>23 CDD...</td>
</tr>
<tr>
<td>18 BBXXYZ...</td>
<td>24 CEE...</td>
</tr>
<tr>
<td>19 BBXXXYZ...</td>
<td>25 CGG...</td>
</tr>
<tr>
<td>20 CBBXYZ...</td>
<td>26 CCH...</td>
</tr>
<tr>
<td>21 CBBBBXYZ...</td>
<td>29 CGH...</td>
</tr>
<tr>
<td>22 CBBXXXYZ...</td>
<td>30 CHH...</td>
</tr>
<tr>
<td>27 CBBBBBBXYZ...</td>
<td>31 DAA...</td>
</tr>
<tr>
<td>28 DAAAA...</td>
<td></td>
</tr>
</tbody>
</table>

Figure 44. Parallel access processing—order of access

With parallel access, DataRefresher always reads from the data set with the lower key field value. After DataRefresher extracts data (equal key field values), DataRefresher reads from the data set with a non-unique key field. The end-of-data in any data set you are joining ends the search.

Joining data using inner and outer loops

If your data sets are not both keyed in the same sequence and you specify to DataRefresher to read data from these data sets and join the extracted data, DataRefresher considers one data set to be the inner data set and the other data set to be the outer data set, and uses an inner processing loop, as shown in Figure 45 on page 198.

Figure 45. Inner processing loop
DataRefresher reads a record from the outer data set, then attempts to match that record in the inner data set. If the inner data set is keyed, DataRefresher reads the record once only. However, if the inner data set is not keyed, DataRefresher must do a sequential search of the inner data set. DataRefresher joins each qualifying entry in the inner data set to the single record from the outer data set. This process may result in one or many rows.

DataRefresher then takes the next qualifying record from the outer data set and again performs an inner processing loop.

The amount of processing involved depends on the structure of the data sets, the amount of data, and any specified selection criteria.

When DataRefresher joins three data sets, it uses two processing loops. DataRefresher finds a qualifying record in the first data set, uses the join field to read the second data set, and uses the second join field to read the third data set.

A keyed inner data set with the join field specified as a unique key means that the data can be obtained from the inner data set with a single read. For a non-keyed, sequential data set DataRefresher must read the entire inner data set every time for each qualifying record in the outer data set.

**Keyed data sets**

For keyed data sets that are not in the same sequence, DataRefresher uses inner loop processing to join them. If the inner data set has a keyed structure and the join is on that key, each time DataRefresher finds a qualifying outer record, the key is used to find any corresponding records in the inner data set with a single read. This can only happen if:

- The data set is an IMS or VSAM KSDS
- SEQFLD=R is specified for the field used to access the inner data set

DataRefresher makes a minimum number of data requests. For example, when processing two keyed data sets, DataRefresher takes the data set with the fewest qualifying records as the outer data set. DataRefresher determines which of the two data sets has the fewest qualifying records by:

- The FREQ keyword on the SEGMENT statement
- The selection criteria on the SELECT statement

If you specify the SEQFLD keyword for the inner data set, DataRefresher reads an outer data set record then an inner data set record searching for a match with the record from the outer data set. DataRefresher stops each pass through the inner data set when a value is returned that is higher than the value read from the outer data set.

If you do not specify the SEQFLD keyword for the inner data set, DataRefresher must read that entire data set for each qualifying record of the outer data set.

**Keyed and non-keyed data sets**

If one data set has a sequence field and the other data set is not sequenced, DataRefresher uses the keyed data set as the inner data set. By doing so, DataRefresher eliminates multiple reads of a sequential data set, and reads the sequential data set only once.
With a combination of keyed and non-keyed data sets, if DataRefresher joins a HDAM database with a HISAM database or a sequential data set, DataRefresher will make the HDAM database the inner data set.

**Non-keyed data sets**
If both data sets are non-keyed, DataRefresher takes the data set with the fewest records as the outer data set.

DataRefresher determines which data sets have the fewest records by:

- The FREQ keyword on the SEGMENT statement
- The selection criteria on the SELECT statement

Non-keyed data sets take more processing time because DataRefresher must read a sequential data set from the beginning each time it starts the inner loop. DataRefresher can not start in the middle of an ESDS because there is no index or other pointer.

If possible, do not join non-sequenced SAM data sets.

**Methods to speed extract processing**

Non-sequenced data sets can reduce performance. If the data is not in a keyed data set, or you do not search on the key, DataRefresher must read the entire data set to find the proper data.

If you join two non-sequenced data sets, or if you join two keyed data sets without specifying the key, DataRefresher will read one of the data sets several times.

**SAM and VSAM data sets**

If you want data from a non-sequenced data set or a non-sequenced field, it is best to have DataRefresher read the entire data set one time and select the requested records on that single pass. Then you could include a sort phase, as part of the JCS, to sort the data.

Joining data sets without causing multiple reads of the same data is complex especially if the data sets are non-sequenced. You could either:

- Convert one sequential data set to a KSDS and use it for extracting
- Or, you could:
  - Extract from one data set only (single pass)
  - Sort the result
  - Load into a KSDS
  - Run an extract from the KSDS and the other data set

In either of these cases, DataRefresher will use the KSDS as the inner data set.
IMS data sets
If the IMS data sets are HDAM or DEDB, their records are in random order. Although their organizations give good online system performance, their organization complicates sequential processing.

Single database: If you only want a small amount of data, you can make DataRefresher retrieve all the records with single calls in the following ways:

- Build and use a secondary index on the search field.
- Code a list of the required records using IN (xxxx, yyyy, zzzz).
- Put the required record keys into a file and join the file to IMS using two views.

If you want to extract all the data in the database, and if you want to load the data in sequence into DB2, it is more efficient to let DataRefresher read the entire database in its physical sequence.

Multiple database join: Because a keyed IMS database can serve as an inner data set, joining multiple IMS databases is not normally a problem. However, problems may arise if DataRefresher has to join two HDAM databases. In most cases, DataRefresher will have to read sequentially through one of the HDAM databases (the outer) to find qualifying records to match to the inner database.

Where DataRefresher reads an entire large IMS database to find only a few qualifying records, the read can be reduced by the methods discussed for a single database.

Choice of outer data set
You might know that a particular choice of outer data set will give better performance because you know that few records will qualify.

If you suspect that DataRefresher has made a less efficient choice, you can tell which choice DataRefresher made by using the DEBUG parameter on the EXTRACT statement and specifying a level of at least 2.

You can influence DataRefresher's choice, if necessary, by using the FREQ parameter to indicate that one file or database is much smaller than the other.

Batching extract requests
The DEM can run more than one extract request at a time, during a single pass of the source database or file. This process of batching extract requests, that is, running extracts together, can save a considerable amount of time and processing resources over running requests one at a time.

See “Batching extract requests” on page 108 for information on batching extract requests.
Tips for better DataRefresher performance

Here are some general guidelines you can follow to enhance DataRefresher performance:

- Always use a keyed field for selective extracts
- Always specify at least the SEQFLD keyword to get the performance benefits of using key sequence fields
- Only specify the SEQFLD keyword for a valid sequence field
- Use a primary key in preference to a secondary key
- Always join data sets on a sequence field
- If possible, do not join non-sequenced SAM data sets
- Anticipate problems with non-sequenced data and plan for them
- Batch extract requests together when possible, to save time and processing resources
- Use equate (‘=’ or ‘IN’) operators if possible when using a numeric key.
Appendix E. DataRefresher dialogs 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 ddbname</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>ISPMLIB</td>
<td>DVR110.DVRMLIBE</td>
<td>Contains DataRefresher dialogs messages.</td>
</tr>
<tr>
<td>ISPSLIB</td>
<td>DVR110.DVRSLIBE</td>
<td>Contains ISPF skeletons that are used when DataRefresher dialogs:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Imbed a request in a JCL file that you want to send</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Build a request from prompt-form request entries</td>
</tr>
<tr>
<td>DVREUADD</td>
<td>DVR110.DVRTADME</td>
<td>Contains the End User Dialogs command table and will contain the master index, end user, and nickname tables built by DataRefresher dialogs during End User Dialogs administration.</td>
</tr>
<tr>
<td>ISPTLIB</td>
<td>DVR110.DVRTLIB</td>
<td>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.</td>
</tr>
<tr>
<td>DVRSTABL</td>
<td>DVR110.DVRTLIB</td>
<td>Contains the object sharing library.</td>
</tr>
</tbody>
</table>

MVS user-specific data set definitions

<table>
<thead>
<tr>
<th>Accessed by ddbname</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.DVR110.DVRDPROF</td>
<td>Will contain ISPF profile information. userid is the user's user ID. Depending on your system configuration, this data set may have another name.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: If the user you are enrolling already has an ISPF profile, do not allocate this data set.</td>
<td>(TRK,(5,5,30)</td>
</tr>
</tbody>
</table>
### VM common file definitions

<table>
<thead>
<tr>
<th>Accessed by ddbname</th>
<th>Default file name</th>
<th>Function/content</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPLLIB</td>
<td>DVRPLIBE MACLIB</td>
<td>Contains the DataRefresher dialogs prompt and HELP panels.</td>
</tr>
<tr>
<td>ISPLLIB</td>
<td>DVRLOAD TXTLIB</td>
<td>Contains the DataRefresher dialogs load library.</td>
</tr>
<tr>
<td>ISPMLIB</td>
<td>DVRMLIBE MACLIB</td>
<td>Contains DataRefresher dialogs messages.</td>
</tr>
<tr>
<td>ISPLLIB</td>
<td>DVSRLIBE MACLIB</td>
<td>Contains ISPF skeletons that are used when DataRefresher dialogs:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Imbed a request in a JCL file that you want to send</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Build a request from prompt-form request entries</td>
</tr>
<tr>
<td>DVREUADD ISPTLIB</td>
<td>DVRTADME MACLIB</td>
<td>Contains the End User Dialogs command table and will contain the master index,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>end user, and nickname tables built by DataRefresher dialogs during End User</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dialogs administration.</td>
</tr>
<tr>
<td>DVRTSTABL</td>
<td>DVRTLIBE MACLIB</td>
<td>Contains the shared objects library.</td>
</tr>
</tbody>
</table>
# VM user-specific file definitions

<table>
<thead>
<tr>
<th>Accessed by ddname</th>
<th>Default file 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 will contain ISPF profile information. Depending on your system configuration, this file may have another name. <strong>Note:</strong> If the user you are enrolling already has an ISPF profile, do not allocate this file.</td>
<td>1 cylinder</td>
</tr>
<tr>
<td>ISPTLIB ISPTABL</td>
<td>DVRTLIBE 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>DVRDJEDI ISPSLIB</td>
<td>DVRJEDIE MACLIB</td>
<td>Will contain 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>Will contain the user's edited extract requests, data description requests, and any JCL files that the user has exported or will import 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.

**CEEPIPI.** 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.

* DB2, DataPropagator, CICS, Language Environment, SAA, AD/Cycle, VTAM, Systems Application Architecture, and SQL/DS are trademarks of the International Business Machines Corporation.

© Copyright IBM Corp. 1986, 1994
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.

cornerstone. 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 (DEDB). 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.

DEDB. 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.

**DRU.** Data Reformat Utility.

**DataRefresher dialogs.** The DataRefresher programs that help users build and send data descriptions (Administrative Dialogs) and extract requests (End User Dialogs and Administrative Dialogs).

**DXTFILE.** A DataRefresher 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 (GDI) exit. (Describes the file organization and selected fields.)

**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.** DataRefresher Program Communication Block.

**DataRefresher Program Communication Block (DXTPCB).** A DataRefresher object stored in the FDTLIB that describes an IMS/VS PCB to DataRefresher.

**DataRefresher Program Specification Block (DXTPSB).** A DataRefresher object stored in the FDTLIB that describes an IMS/VS PSB to DataRefresher.

**DXTPSB.** DataRefresher 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.

**DataRefresher user data type.** Data in a format that DataRefresher does not directly support. A user-written conversion exit is required to convert data from the unsupported format into a format that DataRefresher supports.

**DataRefresher user data type description.** A description of the user-defined format to DataRefresher.

**DXTVIEW.** A DataRefresher 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/VS 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 DataRefresher program that lets a user build and submit extract requests through a series of panels.

**end user table.** A DataRefresher 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/JCS that identifies the program to be run.

**exit address routine.** A user-written routine that, given the name of a GDI exit routine by DataRefresher, returns the current address of that GDI exit routine to DataRefresher.

**exit routine.** A user-written DataRefresher program. DataRefresher 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-sequenced 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-sequenced 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 Language Environment/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 (SYSPUT) 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 (SYSPUT) 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 IXF
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.

RFDLIB. remote file description table library.

root segment. In IMS/VSE, the main segment of a database to which all other segments are related. This is the top of the hierarchy tree.

RSCS. Remote Spooling CommunicationsSubsystem.

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 extract 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
Access Methods Services job
    EXTLIB storage, allocating 94
    FDTLIB storage, allocating 90
Accounting exit routines
    migrating from DXT 37
    overview 84
administration, overview 13
Administrative Dialogs
    CLISTS to set up the dialogs in MVS
        DVRELLGN 160
        DVRELNRL 159, 161
    EXECs to set up the dialogs in VM
        DVREXLIB 167, 170
        DVREXSND 169
        DVREXUSR 167
library, maintaining
    in MVS 164
    in VM 172
migrating from DXT
    in MVS 158
    in VM 165
overview 14
administrator's role 13
Advanced Program-to-Program Communications
    See APPC
ALTER access, RACF 124
APPC
    communications with host MVS system 9
    security 134
APPC software requirements 32
authorization to access data, granting
    See security

B
base product, DataRefrresher 8
batch job, running DEM as 97
batching extract requests
    DXTOUTn data sets 108, 109
    IMS/VS hierarchy 111
    key field 110
    sequence field 110

C
changing a DEM's operating characteristics 117
CLASS Descriptor Table (CDT), RACF 123
CLISTS for starting DataRefrresher dialogs
    DVRELLGN 160
    DVRELNRL 161
commands, using an editor to create 16
communicating with the DEM 112
CONNECT authority for SQL/DS, granting 147
continuous-run mode for the DEM 98
CONTROL access, RACF 124
controlling the DEM 112
CREATE commands 77

D
DAP (Dictionary Access Program)
    administering 135
    batch job as DAP input 136
    data dictionary information, guidelines for 137
    EXECUTE command 136
    JCL for running the DAP 136
    output, reviewing 136
    overview 77
    security for data descriptions 128
    data descriptions
        administering 72
        creating
            CREATE commands 77
            DAP, using 77, 135
    DataRefrresher user data type description
        overview 77
        storage 88
    DVRDEFN access, controlling 124
    DXTFILE description
        overview 76
        storage 88
    DXTPSB description
        overview 76
        PCB description 76, 124
        storage 88
    DXTVIEW description
        overview 77
        storage 89
        security 124, 125
        submitting to the UIM 78
    data dictionary
        DAP, using 135
        EXECUTE command 136
        guidelines for information 137
    Data exit routines
        migrating from DXT 37
        overview 84
    Data Extract
        See DXT
    Data Reformat Utility (DRU) 9
    data set definitions for DataRefrresher dialogs
        MVS common data set 201

© Copyright IBM Corp. 1986, 1994

215
data set definitions for DataRefresher dialogs
(continued)
MVS user-specific data set 201
VM common file 202
VM user-specific file 203
data sets, allocating
dynamically 100
specifying EXTDATA keyword 108
data sources
accessed by DataRefresher 5
protecting from unauthorized use 133
data targets, DataRefresher 7
data-protection schemes
See security
DataPropagator NonRelational, preparing for 86
DataRefresher dialogs
CLISTs to set up the dialogs in MVS
DVRELLGN, editing 160
DVRELNR, calling 161
DVRELNR, editing 159
data set definitions
MVS common data set 201
MVS user-specific data set 201
EXECs to set up the dialogs in VM
DVREXLIB 167, 170
DVREXSND, editing 169
DVREXUSR, editing 167
file definitions
VM common file 202
VM user-specific file 203
library
maintaining in MVS 164
maintaining in VM 172
migrating from DXT
in MVS 157
in VM 165
DataRefresher online commands
DataRefresher features, using with 149
defaults file, creating and modifying 152
environment, setting up 150
libraries, creating 150
overview 15
tailoring 155
task overview 149
DataRefresher OS/2
host installation 32
overview 14
security 134
DataRefresher product family
base product 8
features 8
Date/Time Conversion exit routines, overview 84
DB2 database
security 146
DBCS data 60
debug levels of the DEM 101
decision support systems 3
DEM (Data Extract Manager)
authorization to use 130
batch job, running DEM as 82, 97
data set allocation
dynamically 100
specifying output data sets 108
diagnostic output requirements, determining 101
DXTOUTn data sets for batching requests 108, 109
EXTLIB 37
extract requests
batching 108
priority ranges, setting 99
running 79
FDTLIB 37
initialization commands 102
JCL, necessary information for writing
DEM 102—107
message table, setting up 111
multiple DEMs, running 83
output limits, setting 101
overview 9, 76
protecting from unauthorized use 130
run modes
continuous 98
terminating 99
scheduling 98
security 130
descriptions, data
See data descriptions
diagnostic information, DEM 101
dialogs, DataRefresher
See DataRefresher dialogs
displaying DEM information 113
double-byte character set (DBCS) data 60
DRU (Data Reformat Utility) 9
DVRDEFN resource member class, using
data descriptions
access requirements to perform DataRefresher
operations 125
example 127
implicit access 124
FDTLIB
multiple 132
single 131
resource class
maintaining 126
overview 123
DVRELLGN CLIST for accessing dialogs data
sets 160
DVRELNR CLIST for creating dialogs data sets 159
DVREXLIB EXEC for creating dialogs files 167
DVREXSND EXEC for sending dialogs requests 169
DVREXUSR EXEC for allocating dialogs files 167
DXT
  data sets, using  157
  MACLIBs, using  165
  re-installing  45

DXTFILE description
  access requirements, DVRDEFN  125
  overview  76
  protecting from unauthorized use  124
  storage  88

DXTOUTn data sets  108

DXTPCB
  access requirements, DVRDEFN  125
  overview  76
  protecting from unauthorized use  124

DXTPSB description
  access requirements, DVRDEFN  125
  overview  76
  PCB description  76, 124
  protecting from unauthorized use  124
  storage  88

DXTVIEW description
  overview  77
  storage  89
  dynamically allocating a data set  100

E
  editor, using to create DataRefresher commands  16

End User Dialogs
  CLISTs to set up the dialogs in MVS
    DRELLGN, editing  160
    DRELNRL, calling  161
    DRELNRL, editing  159
  dialogs library
    maintaining in MVS  164
    maintaining in VM  172
  end user table, updating  164

EXECs to set up the dialogs in VM
  DVREXLIB, calling  170
  DVREXLIB, editing  167
  DVREXSND, editing  169
  DVREXUSR, editing  167
  migrating from DXT
    in MVS  157
    in VM  165
  end user table, building
    in MVS  164
    in VM  172
  enrolling dialogs users
    in MVS  161
    in VM  169
  examples of data extracts
    DB2 under MVS  27
    DRU, using under MVS or VM  50
    IBM OS/VS DB/CD Data Dictionary under MVS  27
    IMS library under MVS  28
  examples of data extracts (continued)
    IMS under MVS  26
    SQL/DS under VM  50
  EXECs for starting the DataRefresher dialogs in VM
    DVREXLIB
      calling  170
      editing  167
    DVREXSND  169
    DVREXUSR  167
  EXECUTE command, using to provide DAP input  136
  exit routines
    authorization to use  133
    migrating from DXT  37
    overviews
      Accounting exit routines  84
      Data exit routines  84
      Date/Time Conversion exit routines  84
      Generic Data Interface (GDI) exit routines  85
      Generic Output Interface (GOI) exit routines  86
      Map Capture exit routines  86
      User Data Type exit routines  85
    security  133
  explicit access with DVRDEFN and RACF, granting  124
  EXTLIB (extract request library)
    extract requests, submitting  79
    protecting from unauthorized use  129
    storage requirements
      allocating  93
      estimating  92
      reallocating  95
    extract requests
      batching  108
      creating
        as part of extract process  78
    EXTRACT statement  79
      output limit for the DEM, setting  101
      priority range for the DEM, setting  99
      production  141
      running  79
      storage requirements  92
      SUBMIT command  78
      submitting
        as part of extract process  78
    EXTRACT statement  79

F
  FDTLIB (file description table library)
    data descriptions, submitting  78
    multiple FDTLIBs, using  80, 132
    protecting from unauthorized use  129, 131
    storage
      allocating  89
      estimating  88
      reallocating  91
features, DataRefresher 8
file definitions for DataRefresher dialogs
  MVS common data set 201
  MVS user-specific data set 201
  VM common file 202
  VM user-specific file 203
file descriptions
  See DXTFILE description

G
GDI exit routines 85
General Data Extract feature
  administering
    DAP 135
    DEM operation 71, 97
    Dictionary Access Program 72
    EXTLIB 71
    FDTLIB 71
    Structures Access Program 72
  task overview 71
data descriptions
  building and submitting 76
  DEM operation, preparing 82
  exit routines, preparing 84
  extract requests, building and submitting 75
  installing 47
  overview 8
  preparing to use 75
  repeated extract requests, running 141
security
  DAP data descriptions 128
  data descriptions 124
  DEM 130
  DVRDEFN, using 125
  exit routines 133
  EXTLIB 129
  FDTLIB 129, 131
  overview 17
  source data 133
  UIM 130
Generic Data Interface (GDI) exit routines 85
Generic Output Interface (GOI) exit routines 86
GOI exit routines 86

H
hardware requirements
  DataRefresher MVS/ESA 21
  DataRefresher OS/2 21
  DataRefresher VM/ESA 21
  hierarchy for batching requests, IMS 111

I
implicit access with DVRDEFN and RACF 124
IMS database
  data source 5
  extract processing performance, improving 198
  hierarchy used in batching requests 111
  jobs to run the DEM
    IMS/VS batch job 97, 102, 106
    IMS/VS BMP job 97, 102, 107
  joining multiple databases 199
  key fields, specifying 190
  primary key, extracting data with 192
  secondary indexes, using 192
informational systems 3
INITDEM command 100
installation, DataRefresher
  MVS
    administrative overview 25
    DataRefresher Installation Feature, using 28
    DataRefresher OS/2, preparing host environment 32
    DB2 knowledge required 45
    DXT, re-installing 45
    EXTLIB, creating 37
    FDTLIB, creating 37
    installation task overviews 46
    JES2, preparing to use 45
    libraries, distribution and target 31
    migrating from DXT 37
    PTFs, applying 45
    SMP/E, functions performed by 29
VM
  administrative overview 49
  installation task overviews 55
  libraries, production and distribution 51
  migrating from DXT 53
  PTFs, applying 53
  SQL/DS knowledge required 52
  storage, allocating 51
initialization commands for the DEM 102
ISO formats for date and time data 85

J
Japanese feature 9
JCL
  Dictionary Access Program 136
  information needed to write DEM JCL 102, 103—107
  models in MVS 162
  models in VM 170
JCS
  models in MVS 162
  models in VM 170
JES2 (Job Entry Subsystem 2) 45
Job Entry Subsystem 2 (JES2) 45
joining data
inner and outer loops 196
keyed data sets 197
non-keyed data sets 198
overview 194

K
Kanji Feature
MVS installation 62
overview 11
VM installation 64

M
Map Capture exit routines
DataPropagator NonRelational, preparing for 86
migrating from DXT 37
overview 86
message table for extract request status, setting up 111
migrating from DXT
DataRefresher dialogs 157, 165
exit routines 37
models supplied with DataRefresher 162, 170
MVS operator tasks 112
MVS router table 123

N
National Language Features (NLFs) and DataRefresher
dialogs 57
nickname table, building 164
non-IMS MVS batch job to run the DEM 97, 102—103
non-relational data sources accessed by
DataRefresher 5

O
operational systems 3
operator tasks 112
optional software 22
output, DataRefresher
data targets 7
DEM output
data set allocation by dynamic allocation 100
diagnostic output requirements 101
setting output limits 101
specifying output data sets 108

P
PCB, DXT
See DXTPCB

performance considerations
DataRefresher performance 188
extract processing performance, improving
batching extract requests 108, 199
multiple IMS database join 199
SAM data sets 198
single IMS database 199
VSAM data sets 198
host system performance 187
joining data
inner and outer loops 196
keyed data sets 197
non-keyed data sets 198
overview 194
key fields, specifying
CREATE commands 189
SELECT statement 193
SEQFLD keyword 190
WHERE clause 188
tips for improved DataRefresher performance 200
PERMIT command, RACF 126, 127
physical sequential data set
data source 5
data target 7
polling interval, setting for the DEM 98
priority range, setting extract request 99
program temporary fixes (PTFs), applying
in MVS 45
in VM 53
protecting source data
See security
PSB description
See DXTPSB description
PSDS
See physical sequential data set
PTF
See program temporary fixes, applying

R
RACF 23
access
ALTER 124
CONTROL 124
READ 124, 129—132
UPDATE 124, 129—132
Class Descriptor Table (CDT), adding
DVRDEFN 123
commands
PERMIT 126—127
RALTER 126—127
RDEFINE 126—127
RDELETE 126—127
RLIST 126—127
data descriptions, protecting 124
DEM, protecting 130
RACF (continued)

DVRDEFN resource class
  Class Descriptor Table (CDT), adding to 123
  defining 123
  example 127
  maintaining 126
  MVS router table, adding to 123
  exit routines, protecting 133
  EXTLIB, protecting 129
  FDTLIB, protecting 129, 131
  MVS router table, adding DVRDEFN to 123
  source data, protecting 133
  UIM, protecting 130
RALTER command, RACF 126, 127
RDEFINE command, RACF 126, 127
RDELETE command, RACF 126, 127
READ access, RACF 124, 129—132
  recovering from DEM errors 120
  regular extracts, running 141
  relational data
    data source 5
    data target 7
Relational Data Extract feature
  administering
    data protection, setting up 146
    task overview 145
  data-protection scheme
    CONNECT authority for SQL/DS 147
    overview 17
    relational database views 146
    SQL authorization statements 146
  MVS installation 47
    overview 10
  VM installation 55
  relational database views 146
REM
  See also Relational Data Extract feature
  administrative tasks 145
  overview 10
  repeated extracts, running 141
RLIST command, RACF 126, 127
routines, exit
  See exit routines
run modes for the DEM
  continuous 98
  terminating 99

S
scheduling, DEM
  continuous-run mode 98
  polling interval 98
  terminating-run mode 99
security
  DataRefresher online commands 17
  DataRefresher OS/2 134
security (continued)
  General Data Extract feature
    DAP data descriptions 128
    data descriptions 124, 125
    DEM 130
    DVRDEFN, using 125
    exit routines 133
    EXTLIB 129
    FDTLIB 129, 131
    source data 133
    UIM 130
Relational Data Extract feature
  CONNECT authority for SQL/DS 147
  relational database views 146
  SQL authorization statements 146
security for data 23
SELECT statement
SMP/E
  commands 29
  libraries 31
  sample environment 29
  SMP/E sample environment
    DVR110.DVRSAMPE library 31
    SMPC8S1 30
  zones 30
software requirements 22
DataRefresher MVS/ESA 21
DataRefresher OS/2 21
DataRefresher VM/ESA 21
source data, protecting
  See security
sources, data 5
SQL
  authorization statements 146
  General Data Extract feature 78
SQL/DS databases
  CONNECT authority 147
  data source 5
  data target 7
  security 146
stopping a DEM 115
storage requirements
  DataRefresher
    MVS host system 21
    VM host system 21
  DataRefresher OS/2 21
EXTLIB
  allocating 93
  estimating 92
  reallocating 95
FDTLIB
  allocating 89
  DataRefresher user data type description 88
  DXTFILE description 88
  DXTPSB description 88
  DXTVIEW description 89
  estimating 88
storage requirements (continued) 8
FDLIB (continued) 91
realocating 91
SUBMIT command 78
suppressing WTOR messages 113

T
targets, data 7
task overviews
administration
DataRefresher online commands commands 149
End User Dialogs 164
General Data Extract feature 71
Relational Data Extract feature 145
installation
MVS host system 46
VM/ESA host system 55
terminating-run mode for the DEM 99
types of database 3

U
UCF
See Uppercase Feature
UIM (User Input Manager)
authorization to use 130
data descriptions, submitting 78
extract requests, submitting 78
overview 9
running 75
security 130
UPDATE access, RACF 124, 129—132
Uppercase Feature (UCF)
MVS installation 60
overview 11
VM installation 63
user data type description, DataRefresher
access requirements, DVRDEFN 125
overview 77
protecting from unauthorized use 124
storage 88
User Data Type exit routines 85

V
view
DataRefresher OS/2 14
DXTVIEW description
access requirements, DVRDEFN 125
overview 77
protecting from unauthorized use 124
storage 89
relational database 146
VSAM data set as data source 5

VSE as data target 7

W
write-to-operator-with-reply message 112
WTOR message 112
Communicating Your Comments to IBM

DataRefresher
Version 1
Administration Guide
Publication No. SH19-6995-00

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 to only 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
Administration Guide
Publication No. SH19-6995-00

Overall, how satisfied are you with the information in this book?

<table>
<thead>
<tr>
<th>Overall satisfaction</th>
<th>Very Satisfied</th>
<th>Satisfied</th>
<th>Neutral</th>
<th>Dissatisfied</th>
<th>Very Dissatisfied</th>
</tr>
</thead>
<tbody>
<tr>
<td></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

Address

Company or Organization

Phone No.
IBM Software Solutions,
Information Development (IISL),
2 Burlington Road,
Dublin 4,
Ireland.
**Program Number: 5696-703**

---

### DataRefresher Version 1 Library

<table>
<thead>
<tr>
<th>Code</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH19-5000</td>
<td>Messages and Codes</td>
</tr>
<tr>
<td>LY19-6386</td>
<td>Diagnosis Guide</td>
</tr>
<tr>
<td>GH19-6993</td>
<td>An Introduction</td>
</tr>
<tr>
<td>SH19-6995</td>
<td>Administration Guide</td>
</tr>
<tr>
<td>SH19-6996</td>
<td>MVS and VM User's Guide</td>
</tr>
<tr>
<td>SH19-6997</td>
<td>OS/2 User's Guide</td>
</tr>
<tr>
<td>SH19-6998</td>
<td>Exit Routines</td>
</tr>
<tr>
<td>SH19-6999</td>
<td>Command Reference</td>
</tr>
<tr>
<td>GH19-9994</td>
<td>Licensed Program Specifications</td>
</tr>
</tbody>
</table>

---