

IMS DataPropagator for z/OS



Concepts

Version 3 Release 1

IMS DataPropagator for z/OS



Concepts

Version 3 Release 1

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 67.

This edition applies to Version 3 Release 1 of IMS DataPropagator, 5655-E52, and to any subsequent releases until otherwise indicated in new editions or technical newsletters. This edition is available in softcopy format only.

© **Copyright International Business Machines Corporation 2001, 2007. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	v
About This Information	vii
Terms Used in This Book	vii
How to Use This Book	viii
What You Should Know	viii
Service updates and support information	viii
Receiving information updates automatically	viii
Where to find information	ix
Using LookAt to look up message explanations	ix
Accessibility features	x
How to send your comments	x
Chapter 1. What is IMS DPROP	1
What is data propagation?	1
Key concepts in IMS DPROP	1
Propagation direction	1
Mapping cases	10
Mapping options	11
Exit routines	12
IMS DPROP phases	13
Mapping and design phase	13
Setup phase	16
Propagation phase	18
Maintenance and control phase	30
IMS DPROP functions and components	30
IMS DPROP components	30
MVG, MCE, and MVGU: Creating mapping definitions	31
Capture exit routine	32
Apply program for IMS-to-IMS propagation	33
Apply program for IMS-to-DB2 propagation	33
RUP: Propagating data from IMS to DB2	34
DLU: Loading IMS databases from DB2 tables	35
HUP: Propagating data from DB2 to IMS	35
SCU: Changing the status of propagation requests	35
CCU, AUDU, and Trace: Monitoring propagation and determining problems	36
Sysplex and data sharing	37
How IMS DPROP works in a Sysplex	37
Planning for Sysplex	37
Environmental considerations	38
MQ-ASYNCR propagation environmental considerations	39
LOG-ASYNCR propagation environmental considerations	39
Synchronous propagation environmental considerations	40
User asynchronous propagation environmental considerations	41
Synchronous propagation with asynchronous propagation environmental considerations	41
IMS-to-IMS propagation environmental considerations	42
Chapter 2. Examples using IMS-to-IMS propagation	43
Problem: Systems must be available 24 hours per day, everyday	43
Solution: Keep systems available using IMS-to-IMS replication	43
Problem: A shadow copy is required of some or all primary IMS system databases located at a remote site.	46

	Solution: Use IMS-to-IMS replication features to construct a copy of the	
	databases	46
	Chapter 3. Typical uses of IMS DPROP with IMS-to-DB2 propagation	51
	Coexistence of IMS operational applications and DB2 applications	51
	Coexistence of DL/I update access and SQL read-only access	51
	Coexistence of DL/I read-only access and SQL update access	51
	Coexistence of DL/I update access and SQL update access	52
	Incremental migration of IMS applications	52
	Chapter 4. Trade-offs between synchronous and asynchronous with	
	 IMS-to-DB2 propagation	55
	Advantages and disadvantages of MQ-ASYNCR propagation	55
	Advantages	55
	Disadvantages	56
	Advantages and disadvantages of LOG-ASYNCR propagation	56
	Advantages	56
	Disadvantages	56
	Advantages and disadvantages of synchronous propagation	57
	Advantages	58
	Disadvantages	58
	Chapter 5. How IMS and DB2 relate	59
	Characteristics of IMS databases	59
	IMS database structure	59
	IMS logical databases	60
	IMS secondary indexes	61
	IMS database control blocks	61
	IMS user interface	61
	Characteristics of DB2 databases	61
	DB2 database structure	62
	DB2 indexes	62
	Relating indexes from IMS to DB2	62
	DB2 user interface	63
	IMS and DB2 keys and their relationship	63
	DB2 referential integrity and propagation	65
	Notices	67
	Programming Interface Information	68
	Trademarks	68
	Glossary of Terms and Abbreviations	71
	Bibliography	81
	Index	83

Figures

I	1.	One-way IMS-to-IMS propagation	2
I	2.	IMS-to-IMS MQ-ASYNCR propagation	3
	3.	One-Way IMS-to-DB2 propagation	5
	4.	Overview of MQSeries-based asynchronous propagation	6
	5.	One-way DB2-to-IMS synchronous propagation	9
	6.	Two-way synchronous propagation	10
	7.	Mapping definition phase	15
I	8.	IMS-to-IMS extract using the Image Copy utility and HD Unload utility	17
	9.	IMS-to-DB2 extract and load phase: DataRefresher and the DB2 Load utility	17
	10.	DB2-to-IMS extract and load phase: DLU	18
I	11.	Propagation phase: MQ-ASYNCR asynchronous IMS-to-IMS propagation	19
I	12.	Propagation phase: LOG-ASYNCR Selector used with MQ-ASYNCR propagation	21
	13.	Propagation phase: MQ-ASYNCR asynchronous propagation	22
	14.	Propagation phase: LOG-ASYNCR asynchronous propagation	23
	15.	Propagation phase: IMS-to-DB2 synchronous propagation	24
	16.	Propagation phase: DB2-to-IMS synchronous propagation	25
	17.	Propagation phase: User asynchronous propagation using IMS Asynchronous Data Capture function	27
	18.	Propagation phase: User asynchronous propagation using your Data Capture exit routine	29
	19.	IMS DPROP's Capture Exit routine (EKYMQCAP)	32
I	20.	IMS DPROP Apply program	33
	21.	IMS DPROP's Apply program for DB2	34
I	22.	Preparation of alternate system for receiving IMS transactions	44
I	23.	Service provided during primary system maintenance period	45
I	24.	Switching back to primary system from alternate system	46
I	25.	Overview of process using Capture and Apply exits for outgoing propagation	48
I	26.	Overview of process using Selector program for capture and apply for ongoing propagation	49
	27.	IMS Database Hierarchy	60
	28.	IMS Logical Relationships	61
	29.	Derivation of Concatenated Keys in IMS	64
	30.	Mapping of Keys from IMS to DB2	66

About This Information

IBM® IMS™ DataPropagator for z/OS® (IMS DPROP) is an IMS tool that allows you to maintain consistency between two copies of the same data, where one copy of the data is stored in an IMS Database Manager (IMS DB) database and the other copy is stored in a DB2 database. IMS DPROP is part of the IBM data replication solution.

This user's guide provides instructions for using IMS DPROP. To use the procedures in this user's guide, you must have already installed IMS DPROP using the installation process provided in the Program Directory included with the product.

This information introduces key concepts and capabilities of IMS DPROP. It consists of the following topics:

- Chapter 1, "What is IMS DPROP," on page 1, provides general information about IMS DPROP concepts, functions, and features.
- Chapter 2, "Examples using IMS-to-IMS propagation," on page 43, describes how you can use IMS DPROP for IMS applications that need to access the same data.
- Chapter 3, "Typical uses of IMS DPROP with IMS-to-DB2 propagation," on page 51, describes how you can use IMS DPROP for IMS and DB2 applications that need to access the same data.
- Chapter 4, "Trade-offs between synchronous and asynchronous with IMS-to-DB2 propagation," on page 55, describes and compares asynchronous propagation and synchronous propagation.
- Chapter 5, "How IMS and DB2 relate," on page 59, characterizes IMS and DB2 databases and explains how you relate one to the other in the data propagation environment.

This softcopy book is available only in PDF and BookManager formats. This book is available on the z/OS Software Products Collection Kit, SK3T-4270. You can also get the most current versions of the PDF and BookManager formats by going to the IBM Data Management Tools Web site at www.ibm.com/software/data/db2imstools/library.html.

Terms Used in This Book

The following terms are synonymous in this book:

- *MVS*, *OS/390*, and *z/OS*.
- *File* and *data set*.
- Databases that have been *quiesced* or set to *READONLY* status.

In all cases, these terms refer to:

- Any database you can propagate, except for DEDBs, that is set to READONLY status
- DEDBs that were taken offline with a /DBR command
- *Data Extract (DXT)* and *DataRefresher*.

Unless stated otherwise, these terms refer to DataRefresher Version 1 or higher

IMS DPROP books use the term "child" instead of the term "dependent." For example, IMS DPROP books use the terms "child table" and "child rows" instead of DB2 terms "dependent table" and "dependent rows." The term "child" is used so that terms for IMS and DB2 are similar.

How to Use This Book

This book contains information to help you understand some basic IMS DPROP concepts. The information in this book is applicable to:

- MQSeries-based asynchronous (MQ-ASYNC) propagation
- Log-based asynchronous (LOG-ASYNC) propagation
- Synchronous propagation
- User asynchronous propagation

Any information that applies to only one type of propagation is indicated as follows:

- Sentences and phrases are specified explicitly, for example:

If you are implementing synchronous propagation,...

- List items, paragraphs and sections are marked by a symbol in the margin, for example:

With one-way DB2-to-IMS propagation, IMS DPROP propagates database updates from DB2 tables to IMS databases.

Margin symbols are:

Q MQ-ASYNC propagation only

A Asynchronous propagation only (particularly LOG-ASYNC asynchronous)

S Synchronous propagation only

User asynchronous is not marked by a special margin symbol but is indicated by sentences, phrases, and headings.

These margin symbols are used throughout the IMS DPROP library.

What You Should Know

This information assumes that you understand what data propagation is and the business reasons for propagating data. Information on these topics is in the manual titled: *An Introduction*.

This information also assumes that you understand IMS, DB2, and DataRefresher concepts and functions.

Service updates and support information

To find service updates and support information, including software fix packs, PTFs, Frequently Asked Question (FAQs), technical notes, troubleshooting information, and downloads, refer to the following Web page:

www.ibm.com/software/data/db2imstools/support.html

Receiving information updates automatically

By registering with the IBM My Support service, you can automatically receive a weekly e-mail that notifies you when new DCF documents are released, when existing product documentation is updated, and when new product documentation is available. You can customize the service so that you receive information about only those IBM products that you specify.

To register with the My Support service:

1. Go to <http://www.ibm.com/support/mysupport>
2. Enter your IBM ID and password, or create one by clicking **register now**.
3. When the My Support page is displayed, click **add products** to select those products that you want to receive information updates about. The DB2® and IMS Tools category is located under **Software -> Data and Information Management -> Database Tools & Utilities**.
4. Click **Subscribe to email** to specify the types of updates that you would like to receive.
5. Click **Update** to save your profile.

Where to find information

The DB2 and IMS Tools Library Web page provides current product documentation that you can view, print, and download. To locate publications with the most up-to-date information, refer to the following Web page:

www.ibm.com/software/data/db2imstools/library.html

IBM Redbooks™ that cover DB2 and IMS Tools are available from the following Web page:

www.ibm.com/software/data/db2imstools/support.html

Using LookAt to look up message explanations

LookAt is an online facility that lets you look up explanations for most of the IBM messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can use LookAt from the following locations to find IBM message explanations for z/OS elements and features, z/VM®, VSE/ESA™, and Clusters for AIX® and Linux®:

- The Internet. You can access IBM message explanations directly from the LookAt Web site at <http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/>.
- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations, using LookAt from a TSO/E command line (for example, TSO/E prompt, ISPF, or z/OS UNIX® System Services running OMVS).
- Your Microsoft® Windows® workstation. You can install code to access IBM message explanations on the z/OS Collection (SK3T-4269) using LookAt from a Microsoft Windows command prompt (also known as the DOS command line).
- Your wireless handheld device. You can use the LookAt Mobile Edition with a handheld device that has wireless access and an Internet browser (for example, Internet Explorer for Pocket PCs, Blazer, or Eudora™ for Palm OS, or Opera for Linux handheld devices). Link to the LookAt Mobile Edition from the LookAt Web site.

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from a disk on your z/OS Collection (SK3T-4269) or from the LookAt Web site (click **Download**, and select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully. The major accessibility features in this product enable users to:

- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.
- Customize display attributes such as color, contrast, and font size.
- Operate specific or equivalent features by using only the keyboard. Refer to the following publications for information about accessing ISPF interfaces:
 - *z/OS ISPF User's Guide, Volume 1*, SC34-4822
 - *z/OS TSO/E Primer*, SA22-7787
 - *z/OS TSO/E User's Guide*, SA22-7794

These guides describe how to use ISPF, including the use of keyboard shortcuts or function keys (PF keys), include the default settings for the PF keys, and explain how to modify their functions.

People with limited vision who use screen reader software might find the following feature requires particular attention:

Pop-up windows

This product uses ISPF function that produces pop-up windows for some tasks. The pop-up and its frame are just text that overlays the underlying information on the displayed panel. The frame of such a pop-up is not usually recognized as such by screen reader software, so you may need to gain some familiarity with reading such panels before the information becomes meaningful. Alternatively, you can display the pop-up window on a full screen by using the RESIZE command or function key.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this information or any other IMS DataPropagator for z/OS documentation, use either of the following options:

- Use the online reader comment form, which is located at:
www.ibm.com/software/data/rcf/
- Send your comments by e-mail to comments@us.ibm.com. Be sure to include the name, part number and version of the book and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

Chapter 1. What is IMS DPROP

What is data propagation?

Data propagation consists of taking the changes made to one set of databases and applying them to another set of databases.

The IMS DataPropagator (IMS DPROP) propagates updated data:

- From IMS to IMS
- From IMS to DB2 databases
- From DB2 to IMS databases

Propagation allows you to maintain consistency between two copies of the same data; one copy of the data is stored in IMS databases and the other copy is stored in IMS databases or in DB2 tables. IMS DPROP allows IMS and DB2 applications to coexist and access copies of the same data. With IMS DPROP:

- IMS applications can access the IMS data copy through the IMS DL/I call interface.
- DB2 applications can access the DB2 copy of the same data through the DB2 SQL interface.

The process of propagation is transparent to the application program which is performing the updates to the propagation source data.

Key concepts in IMS DPROP

Data propagation involves applying changes made to one set of data to the copy of that data in another database system, keeping two sets of data in different database systems consistent.

This section describes key concepts you need to understand when propagating data using IMS DPROP:

- Propagation direction
- Propagation modes
- Mapping cases and options
- Exit routines

Propagation direction

IMS DPROP propagates data using four methods:

- One-way IMS-to-IMS
- One-way IMS-to-DB2
- One-way DB2-to-IMS
- Two-way

One-way IMS-to-IMS propagation

With one-way IMS-to-IMS propagation, IMS DPROP propagates database updates from source IMS databases to target IMS databases. One-way IMS-to-IMS propagation involves maintaining an IMS database as a master copy of the data and having a duplicate copy of the data maintained in another IMS database.

With one-way IMS-to-IMS propagation, you first make a copy of the source IMS data in another (target) copy of the IMS data. For example: Run the HD Unload utility to obtain a copy of the IMS source database and use the unloaded file with the HD Reload utility to load the IMS target database. Then, when changes are

made to the IMS source data, IMS DPROP applies (propagates) these changes to the IMS target data.

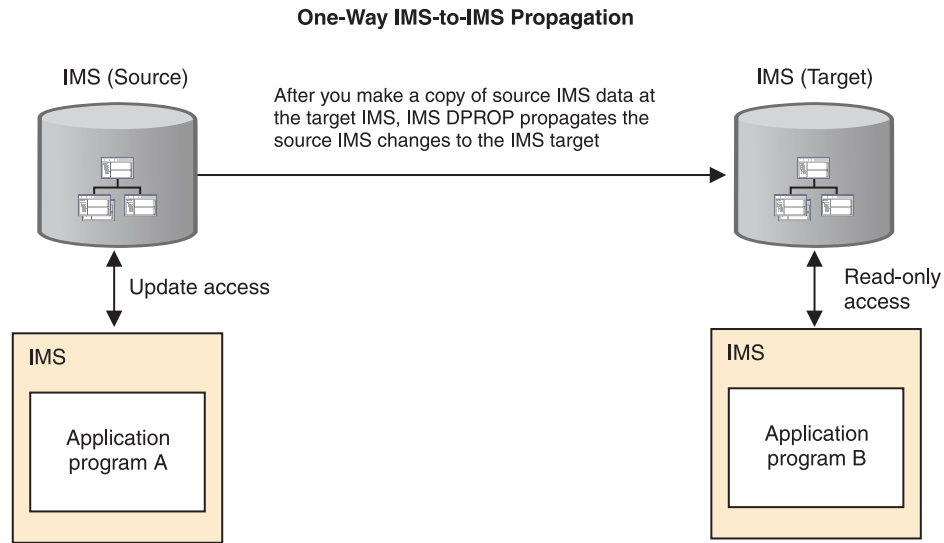


Figure 1. One-way IMS-to-IMS propagation

You can perform one-way IMS-to-IMS propagation asynchronously, in which case IMS DPROP propagates to the target IMS database, the changes that have been made to the source IMS database. You can perform this type of propagation using either:

- MQSeries-based (MQ-ASYNC) asynchronous propagation, or
- A subset of log-based (LOG-ASYNC) asynchronous propagation features with MQSeries-based (MQ-ASYNC) asynchronous propagation

MQ-ASYNC propagation: In pure MQ-ASYNC propagation mode, IMS DPROP propagates changes made to an IMS source database to an IMS target database copy using the MQSeries product as the bridge between the two IMS databases. This mode of propagation provides near real-time updates to the IMS target with minimal impact to the IMS source application which originally made the updates.

You can use IMS-to-IMS MQ-ASYNC propagation when:

- You have a requirement for a very current target system IMS copy of your IMS source data, and when the time difference between the IMS source update and the propagation to the IMS target database needs to be small.
- You have a requirement to perform maintenance activities on your normally active IMS databases, and you want this IMS data to be available during that maintenance period. In such a scenario, you would allow normal IMS application updates to a duplicate copy of your master data and propagate the changes made to this IMS copy back to the master database after maintenance activities are complete.

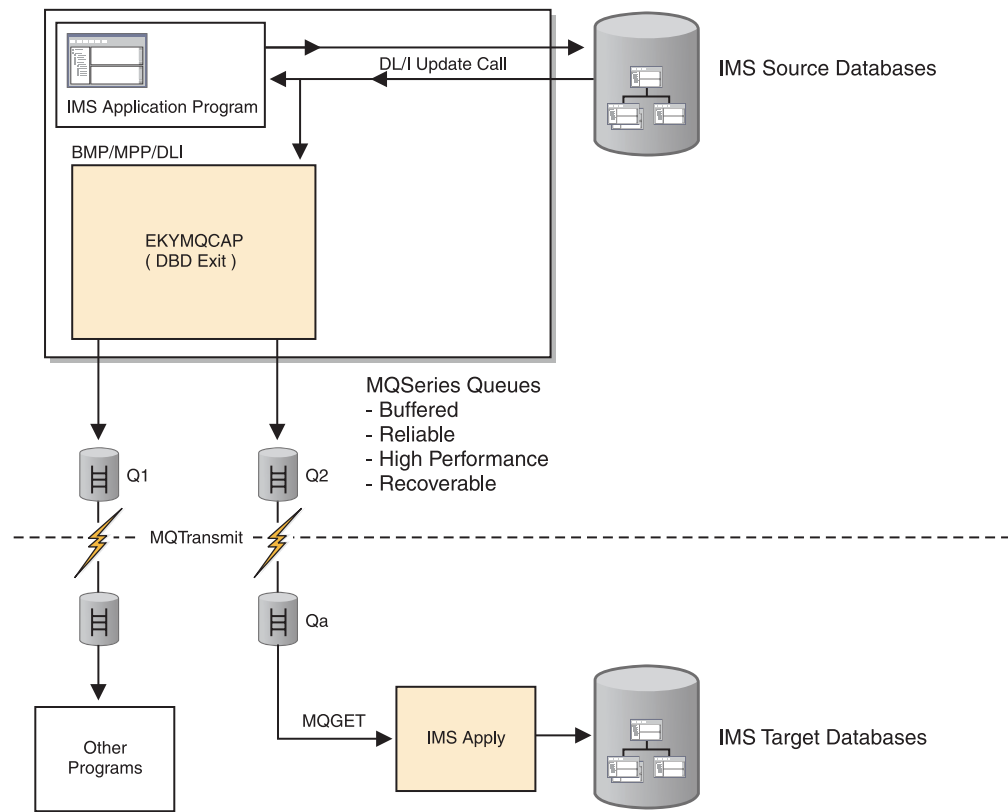


Figure 2. IMS-to-IMS MQ-ASYNC propagation

The following list describes the process shown in Figure 2:

1. When an IMS application changes an IMS segment type to be propagated, DPROPC captures the update using a Capture exit routine. This “capture” happens during the unit-of-work (UOW) of the IMS application. Control is passed to the Capture exit routine after the update to the segment is made.
2. The Capture exit routine builds an MQSeries message containing the information about the changed segment and places the message on an MQSeries message queue (labeled Q2 in Figure 2). The Capture exit routine can place the message on multiple queues (for example those labeled both Q1 and Q2), as shown in Figure 2.
3. If all processing performs with no problems (update to the segment, building of the message, and placing the message on the MQSeries queue), the update to the segment is committed. The IMS application’s UOW is now complete.
4. After the segment change is committed, MQSeries transmits the message to another MQSeries queue (labeled Qa in Figure 2) on the system where the IMS Apply program is running. This second system is called the target system.
The changed IMS data flows from the Capture component to the IMS Apply program occurrences as part of one or more *propagation data streams* (PRSTREAMs). All changed data contained in one PRSTREAM is transmitted by means of the same MQSeries queues and is applied in First-In First-Out (FIFO) sequence by the same occurrence of the IMS Apply program.
5. The IMS DPROPC IMS Apply program retrieves the message from queue Qa and updates the IMS target database.

With MQ-ASYNC IMS-to-IMS propagation, the IMS target database can reflect changes to the IMS source database rapidly. When conditions are optimal, the IMS target copy can reflect IMS source updates in near-real-time.

LOG-ASYNC with MQ-ASYNC propagation: In this propagation mode, IMS DPROP propagates to an IMS target database copy all changes that were made to an IMS source database. This is accomplished by using a subset of features from LOG-ASYNC IMS DPROP and MQ-ASYNC IMS DPROP components. Usually, you would use this type of "hybrid" propagation when any of the following conditions is true:

- Your IMS source database updates are performed outside of the direct control of IMS; for example, when your online IMS update transactions are initiated through CICS/DBCTL.
- There is no requirement for near real-time source-to-target propagation.
- The performance impact on existing operational applications must be minimized.

In this propagation mode, IMS DPROP propagates changes to an IMS target database target that were made to an IMS source database. This is accomplished by selecting source database changes initially captured from IMS log records. The changed source data captured within the log records is then subsequently propagated to the IMS target system using the same MQ-ASYNC components providing for normal MQ-ASYNC IMS-to-IMS propagation described above.

One-way IMS-to-DB2 propagation

With one-way IMS-to-DB2 propagation, IMS DPROP propagates database updates from IMS databases to DB2 tables. One-way IMS-to-DB2 propagation involves maintaining the IMS database as the master copy of the data and having a subordinate (complete or partial) copy of the data in DB2.

With one-way IMS-to-DB2 propagation you first make a copy of IMS data in DB2. (For example, you can use your own program to extract data from IMS databases and load the data into DB2 tables.) Then, when changes are made to the IMS data, IMS DPROP applies (propagates) these changes to DB2.

When you implement one-way IMS-to-DB2 propagation, you can develop programs that access the propagated DB2 data. Unless you implement two-way synchronous propagation, the DB2 programs you develop typically use read-only SQL statements, because updates to the DB2 copy are not propagated back to the IMS copy.

As shown in Figure 3 on page 5, one-way IMS-to-DB2 propagation allows the coexistence of applications with:

- DL/I update access to the propagated IMS data.
- SQL read-only access to the propagated DB2 data copy. You can also use the DB2 data for decision support purposes.

IMS DPROP lets you propagate changes from *selected* IMS databases, segments, and fields so you can specify what data is propagated down to the level of fields within an IMS segment.

One-Way IMS-to-DB2 Propagation

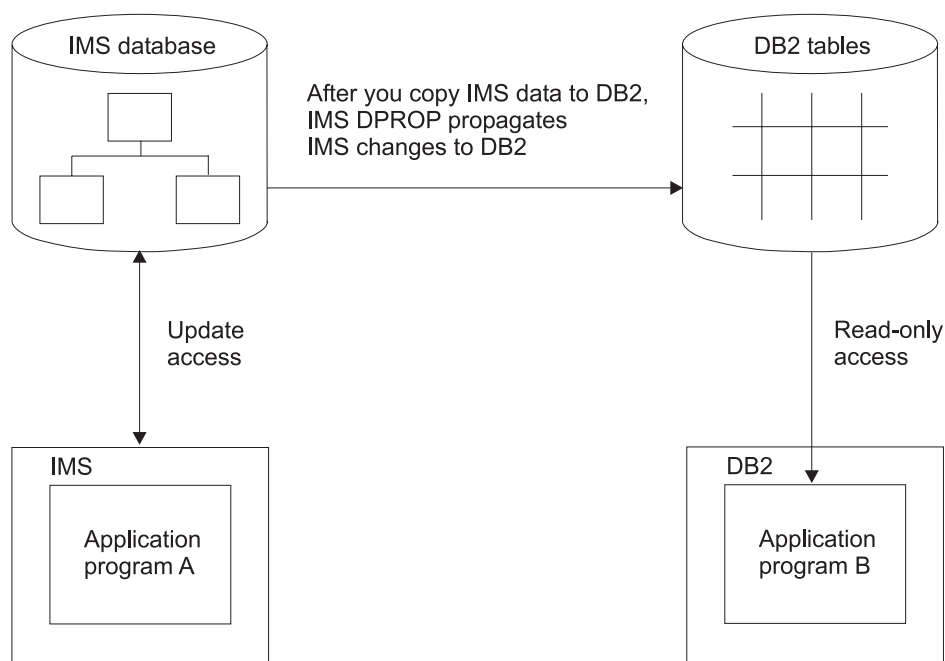


Figure 3. One-Way IMS-to-DB2 propagation

You can perform one-way IMS-to-DB2 propagation:

- Asynchronously, using either:
 - MQSeries-based (MQ-ASYNC) asynchronous propagation
 - Log-based (LOG-ASYNC) asynchronous propagation
 - User asynchronous propagation
- Synchronously
- Both synchronously and asynchronously

One-way IMS-to-DB2 asynchronous propagation

With asynchronous propagation, IMS DPROT propagates the changes that have been made to the IMS database to the DB2 copy at a later time, not within the same unit of work (UOW) as your IMS application's update call. In this way, target DB2 tables are periodically updated with the point-in-time state of a source IMS database.

MQ-ASYNC propagation: In MQ-ASYNC propagation mode, IMS DPROT propagates changes made to an IMS database to a DB2 copy of the IMS data using the MQSeries product as the bridge between the two database managers. This mode of propagation provides near real-time updates to DB2 with minimal impact to the IMS application that originally made the updates. MQ-ASYNC propagation supports one-way IMS-to-DB2 propagation.

You usually use MQ-ASYNC propagation when:

- You have a requirement for data to be very current on the target system (the time difference between the IMS update and the propagation to DB2 needs to be small).

- Your DB2 data is used for an information warehouse or in other decision-support applications, such as queries or the generation of reports
- The performance impact on existing operational applications must be minimized.
- IMS and DB2 are running on different MVS images.

Figure 4 illustrates how MQ-ASYNC propagation works.

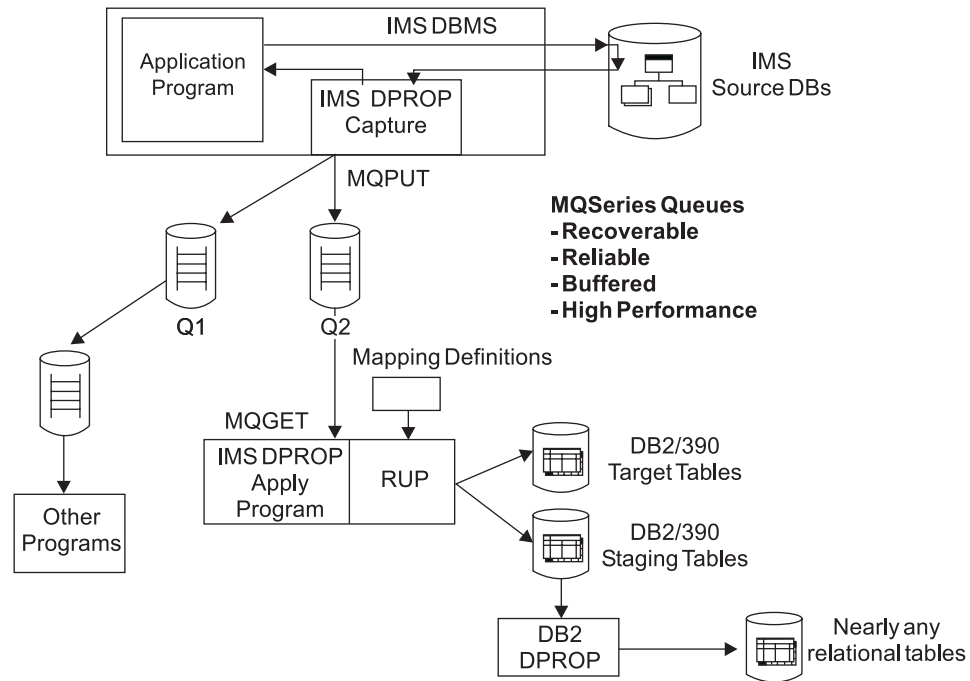


Figure 4. Overview of MQSeries-based asynchronous propagation

The following list describes Figure 4:

1. When an IMS application changes an IMS segment type that is to be propagated, IMS DPROT captures the update using a Capture exit routine. This “capture” happens during the unit of work of the IMS application. Control is passed to the Capture exit routine after the update to the segment is made.
2. The Capture exit routine builds an MQSeries message containing the information about the changed segment and places this message on an MQSeries message queue (labeled Q2 in Figure 4). The Capture exit routine can place the message on multiple queues, as shown in Figure 4.
3. If all the processing performed with no problems (the update to the segment, the building of the message and the placing of the message on the MQSeries queue), the update to the segment is committed. At this point in time, the IMS application’s unit-of-work is complete.
4. The IMS DPROT Apply program retrieves the message from queue Q2 and passes it to the IMS DPROT Relational Update Program (RUP).
5. The RUP queries the mapping definitions that describe how the IMS change data “maps” to the target DB2 tables.
6. Using the information from the mapping definitions, the RUP updates the target DB2 tables.
7. If the target DB2 tables are staging tables, the DB2 DataPropagator product can propagate the change data to almost any other relational database.

With MQ-ASYNC propagation, the DB2 tables can reflect changes to the IMS database much faster than using LOG-ASYNC propagation. If conditions are optimal, the DB2 copy can reflect the IMS updates in near-real time.

LOG-ASYNC propagation: In LOG-ASYNC propagation mode, IMS DPROP supports only one-way IMS-to-DB2 propagation. Neither DB2-to-IMS nor two-way propagation is supported.

You usually use LOG-ASYNC propagation when:

- Your DB2 data is used for an information warehouse or in other decision-support applications, such as queries or the generation of reports, that do not need updates in real time.
- The performance impact on existing operational applications must be minimized.
- IMS and DB2 are running on different MVS images.

When you update an IMS segment type that is to be propagated, IMS captures the changed segment occurrences and writes them to the IMS log. You can modify the IMS archive exit JCL so that it filters out propagation log records (log records associated with data specified for propagation) from IMS Online data sets (OLDS) and writes them to a separate data set called the Changed Data Capture data set (CDCDS). The following LOG-ASYNC propagation components control the propagation of the propagation log records from the IMS databases to the DB2 tables:

Selector

Runs on the same MVS image as the IMS Database Recovery Control (DBRC) and uses the DBRC to find log files. The Selector extracts propagation log records from either the IMS logs, the CDCDSs, or both and writes them to sequential data sets called Propagation Request data sets (PRDS), ensuring that no gaps or overlaps exist.

Receiver

Runs on the same MVS image as the DB2 tables being propagated to. It retrieves the changes from the PRDSs and applies them to the DB2 tables.

If you encounter any problems on the DB2 side, you do not need to stop IMS:

- If the Selector and the Receiver are running on the same MVS image or on local MVS images (MVS images with shared DASD resources), the Receiver can retrieve the changes directly.
- If the Selector and the Receiver are running on remote MVS images, you must arrange to transport the changes to the Receiver MVS image.

With LOG-ASYNC propagation, data is always current to a specified point in time.

One-way IMS-to-DB2 synchronous propagation

With synchronous propagation, updates are propagated immediately, within the same UOW as your application's update call.

In synchronous mode, IMS DPROP supports both one-way and two-way synchronous propagation.

You usually use synchronous propagation if you are using both data copies in operational applications, such as:

- Manufacturing, for example, checking the inventory of a part or updating the inventory of a part

- Banking, for example, updating a customer's address

Synchronous propagation is appropriate for applications in which both copies of the data must be up-to-date.

With synchronous propagation, IMS and DB2 must run under the same MVS image. If the propagation of an update fails, backout is done on both the IMS and DB2 side to keep data consistent. If IMS DPROP encounters problems when propagating updates (for example, if the target copy is not available), you must stop applications that update data that is subject to propagation or, for critical applications, deactivate propagation.

One-way IMS-to-DB2 user asynchronous propagation

Use user asynchronous propagation only if you implemented this type of propagation in a previous release of IMS DPROP. User asynchronous propagation is similar to asynchronous propagation, except that you must write programs that complement the functions of IMS and IMS DPROP. When there is an update to an IMS segment type that is to be propagated, IMS captures the changed segment occurrences. What happens next depends on the implementation chosen by your installation. The changed data can be saved on:

- The IMS log by the IMS Asynchronous Data Capture function.
- A database, MVS flat files, or the IMS log by a user-written Data Capture exit routine. You can also use user-written Data Capture exits to store the changed data in IMS output messages and send these messages to other programs, possibly over an IMS MSC or ISC link to another location.

Later, user-written programs can retrieve the changed data and invoke IMS DPROP to apply the changes to DB2 tables.

Implementing user asynchronous propagation is somewhat complex because of the programs you must write.

One-way DB2-to-IMS synchronous propagation

With one-way DB2-to-IMS synchronous propagation, IMS DPROP propagates database updates from DB2 tables to IMS databases. One-way DB2-to-IMS synchronous propagation involves maintaining the DB2 tables as the master copy of the data and having a subordinate (complete or partial) copy of the data in IMS.

When performing one-way DB2-to-IMS synchronous propagation, you usually make a copy of DB2 data in IMS. (For example, you can use a IMS DPROP utility or your own program to extract data from the DB2 tables and load data into the IMS database.) When you implement one-way DB2-to-IMS synchronous propagation, your IMS programs can use the propagated IMS copy of your DB2 tables. Then, when changes are made to the DB2 data, IMS DPROP propagates these changes to IMS.

Unless you implement two-way synchronous propagation, your IMS programs usually use read-only DL/I calls, because updates made to the IMS copy are not propagated back to the DB2 copy. See Figure 5 on page 9 for an illustration of this propagation direction.

As shown in Figure 5 on page 9, one-way DB2-to-IMS synchronous propagation allows the coexistence of applications with:

- SQL update access to the propagated DB2 data
- DL/I read-only access to the propagated IMS data copy

IMS DPROP lets you propagate changes from *selected* DB2 tables and columns. You can specify what data is propagated down to the level of columns within a DB2 table.

One-Way DB2-to-IMS Synchronous Propagation

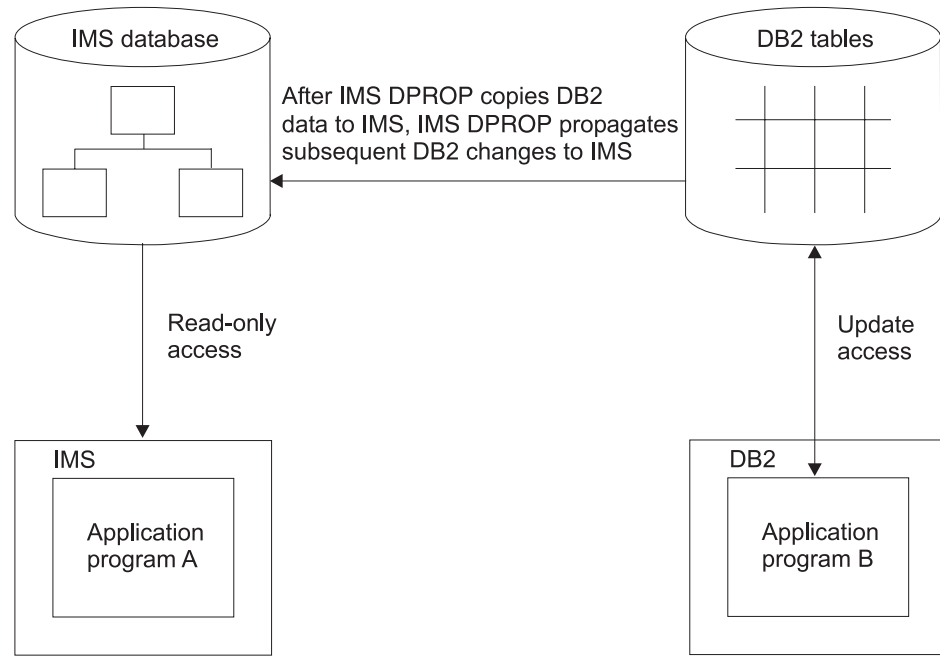


Figure 5. One-way DB2-to-IMS synchronous propagation

You can perform one-way DB2-to-IMS propagation only in synchronous mode, as explained in “Propagation direction ” on page 1.

Two-way synchronous propagation

Two-way synchronous propagation propagates data from IMS to DB2 and from DB2 to IMS.

With two-way synchronous propagation, you first make a copy of IMS data in DB2, or DB2 data in IMS. Then, when changes are made to the IMS or DB2 data, IMS DPROP applies these changes to the other database as well. Neither the IMS nor DB2 copy of the data is considered the master copy; there is no master copy of the data.

As shown in Figure 6 on page 10, two-way synchronous propagation allows the coexistence of applications with:

- DL/I update access to the propagated IMS data
- SQL update access to the propagated DB2 data copy

When compared to one-way propagation, two-way synchronous propagation offers greater coexistence between programs using DL/I calls and programs using SQL statements to access the same data. However, two-way synchronous propagation has a substantial disadvantage. If the two copies of the data become inconsistent (for example because of operational errors) it is difficult to resynchronize the two

copies, because there is no master copy. IMS DPROP does not know whether it is the IMS or DB2 copy that is accurate. In such situations, you must determine which copy is accurate and how the two copies can be resynchronized in your application environment. If you determine that your application environment can handle situations with no master copy of the data, it is recommended that you develop and test procedures for resolving inconsistencies.

Two-Way Synchronous Propagation

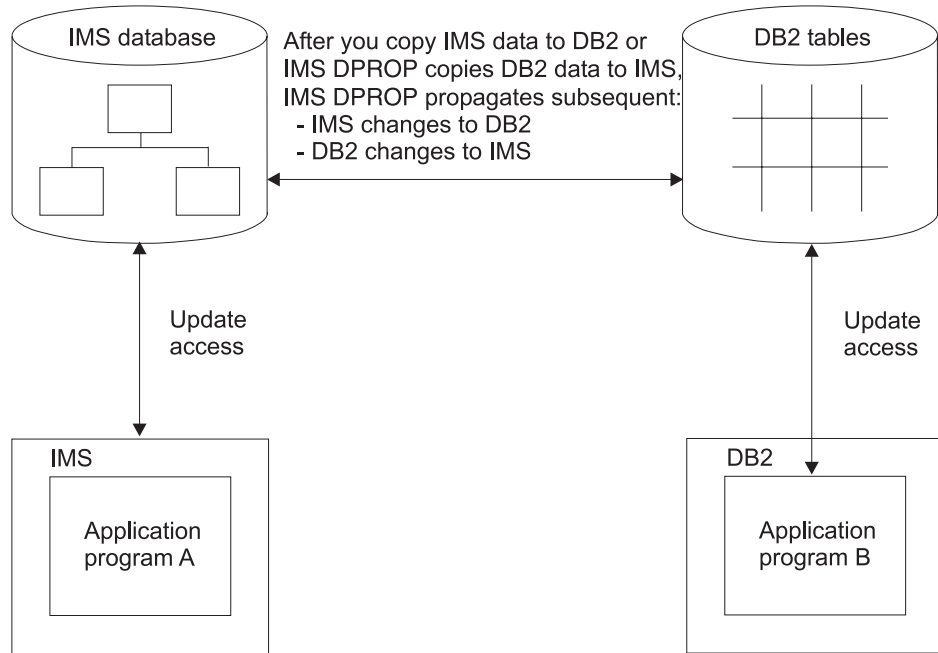


Figure 6. Two-way synchronous propagation

Two-way synchronous propagation is sometimes useful as an intermediate step in migrating IMS applications. For more information on this topic, see “Incremental migration of IMS applications ” on page 52.

You can perform two-way propagation only in synchronous mode, as explained in “Propagation direction ” on page 1

Mapping cases

When data is propagated between IMS and DB2, data from keys and fields in an IMS segment must be associated with, or mapped to, rows and columns in a DB2 table. A *mapping case* defines how data is mapped between an IMS segment and a DB2 table. IMS DPROP provides three different generalized mapping cases:

Case 1

Used to propagate one single segment type occurrence, with the keys of the physical parent and ancestors up to the root, to or from a row in a single DB2 table.

Case 2

Used to propagate one single segment type occurrence, with the keys of the physical parent and ancestors up to the root, plus data from one or

more immediately subordinate segment types to or from a row in a single DB2 table. Only one or zero occurrences of each immediately subordinate segment type can exist for each parent.

Case 3

Used to propagate embedded structures contained in IMS segments. An embedded structure is a group of fields; a typical example of an embedded structure is a repeating group of fields.

You can propagate:

- Each embedded structure to or from a different DB2 table. Mapping case 3 maps each occurrence of an embedded structure to or from a row in the corresponding DB2 table. The mapping can include:
 - Any embedded structure
 - The key of the segment
 - The keys of the physical parent and ancestors up to the root
- The remaining data in the segment (the fields not located in the embedded structure) to or from another DB2 table.

Mapping cases 1, 2, and 3 support a standard set of data conversions without requiring the use of field or segment level exits.

IMS DPROP also supports the use of *user mapping cases* using a Propagation exit routine you write as described in “Propagation exit routine” on page 13.

You specify which mapping case to use in a propagation request. You define propagation requests for each segment type or table that is to be propagated.

Mapping options

IMS DPROP supports the following mapping options for use with the generalized mapping cases:

WHERE clause

Specifies the conditions in which a changed IMS segment occurrence propagates to DB2, for example, when a specific combination of field values exists.

By defining multiple propagation requests with different WHERE clauses, you can propagate the same segment type to different tables. This use of the WHERE clause is useful when a segment contains different kinds of data, for example, redefined data.

Use the WHERE clause with mapping cases 1 and 2.

PATH data

Lets you include non-key fields of the physical parent and physical ancestors in the mapping of a segment.

IMS DPROP provides the following types of PATH data:

- PATH data that includes fields in the parent/ancestors that *can* change their value. Use this type of PATH data to intentionally denormalize data to improve performance of DB2 queries of the propagated tables. This use is supported for only one-way IMS-to-DB2 propagation.
- PATH data that includes only fields from the parent/ancestors that *cannot* change their value. This type of PATH data better supports propagation of IMS parent segments that do not have a unique IMS key field but can nevertheless be uniquely identified through a combination of:
 - Non-key fields that do not change their value

- The non-unique key field, if any

This second use of PATH data does not result in denormalization and is supported for both one- and two-way propagation.

Use the PATH data with mapping cases 1, 2, and 3.

Exit routines

Notice

This section contains general-use programming interface and associated guidance information.

IMS DPROP's generalized mapping (mapping case 1, 2, and 3 with the WHERE clause and PATH data option) should meet most of your needs. If not, you can complement the generalized mapping logic by writing optional Segment and Field exit routines.

During IMS-to-DB2 mapping, IMS DPROP calls the Segment and Field exit routines *before* performing its own mapping and data conversion. During DB2-to-IMS mapping, IMS DPROP calls the Field and Segment exit routines *after* performing its own mapping and data conversions. During DB2-to-IMS mapping, the editing performed by your exit routines is the reverse of the editing performed during IMS-to-DB2 mapping.

If Segment and Field exit routines still do not meet your needs, for some segments you can use your own mapping and propagation logic by writing a Propagation exit routine. Writing a Propagation exit routine is more complex than writing a Segment or Field exit routine, because it must provide all the logic for data mapping, field conversion, and propagation.

Refer to the *Customization Guide* for details of writing exit routines.

Segment exit routine

Segment exit routines:

- Map IMS fields with variable start positions to fixed positions
- Clean up data where IMS data was stored in redefined areas
- Change the content of data stored in an IMS segment or restructure the segment before it is mapped and propagated to the DB2 table
- Support some mapping case 3 propagation requests, for example, those used to perform DB2-to-IMS synchronous propagation
- Selectively suppress the propagation of some changes

A Segment exit routine cannot change the format or content of keys, concatenated keys, or any field mapping to the primary DB2 key.

Field exit routine

Field exit routines:

- Convert IMS fields with formats that are not supported by IMS DPROP
- Allow additional IMS date and time formats. Without a Field exit routine, IMS DPROP supports only one DB2 date and time format at a time: ISO, USA, EUR, or JIS.
- Convert some values in an IMS field to a DB2 null value

- Change the content of an IMS key field

Propagation exit routine

Use Propagation exit routines to replace IMS DPROPS generalized mapping logic when you need specialized or complex mapping. Propagation exit routines work with the following IMS DPROPS service functions:

- Tracing
- Processing standardized errors
- Generating an audit trail
- Changing the status of a propagation requests

IMS DPROPS phases

Propagation with IMS DPROPS consists of four phases:

- Mapping and design
- Setup
- Propagation
- Maintenance and control

This section describes key events that occur during each phase.

Mapping and design phase

In the mapping and design phase, you define how you want to implement propagation. This section gives an overview of how to:

- Identify the propagated data to IMS DPROPS and to:
 - IMS, for IMS-to-IMS propagation
 - IMS, for IMS-to-DB2 propagation
 - DB2, for DB2-to-IMS synchronous propagation
- Specify the mapping (only applicable for IMS-to-DB2 and DB2-to-IMS propagation)

Identifying the data to be propagated

To define data to be propagated:

IMS-to-IMS propagation

Identify the databases subject to propagation to IMS by making the following minor modifications to the database definitions (DBDs) of the associated IMS source databases:

- When using only MQ-ASYNCH features for IMS-to-IMS propagation, specify in the DBD that IMS is to invoke the IMS DPROPS Capture exit routine (EKYMQCAP) when changes are to be made to IMS data which is to be propagated.
- When using LOG-ASYNCH to initially capture IMS changes for IMS-to-IMS propagation, specify in the DBD that IMS is to write IMS source data changes to the IMS log by means of the IMS log exit.

IMS-to-DB2 propagation

Identify the segments subject to propagation to IMS by making the following minor modifications to the database definitions (DBDs) of the associated IMS databases:

- For MQ-ASYNCH propagation, specify in the DBD that IMS is to call the IMS DPROPS Capture exit routine (EKYMQCAP) after the IMS data is updated.
- For LOG-ASYNCH propagation, specify in the DBD that IMS is to write the changes to the IMS log.

- For synchronous propagation, specify in the DBD that IMS is to invoke IMS DPROP when segments subject to propagation are changed.
- For user asynchronous propagation, specify in the DBD that IMS is to perform one of the following actions when segments subject to propagation are changed:
 - Write the changes to the IMS log.
 - Invoke a Data Capture exit routine (that you provide). The Data Capture exit routine saves the changed data in the IMS log or a database.

DB2-to-IMS synchronous propagation

Use the DB2 CREATE TABLE or ALTER TABLE statements to identify the tables subject to propagation. Use of these statements (along with other specifications) invokes IMS DPROP when changes are made to these tables.

Two-way synchronous propagation

Requires you to identify both the segments and tables being propagated.

Specifying the mapping

For IMS-to-IMS propagation, normally the IMS target database duplicates the structure of the IMS source database. Specific data mapping is not applicable with this propagation mode.

Specify in IMS DPROP the IMS segments and DB2 tables to be mapped and how the mapping is to be done. As shown in Figure 7 on page 15, you can specify mapping information by:

- Using a combination of DXT or DataRefresher and IMS DPROP.
- Entering mapping information directly into the Mapping Verification and Generation (MVG) DB2 input tables.
Use the IMS DPROP TSO ISPF application or a QMF or SQL application that you write to put mapping information directly into the MVG.

See “MVG, MCE, and MVGU: Creating mapping definitions” on page 31 for more information on the components you use to specify mapping.

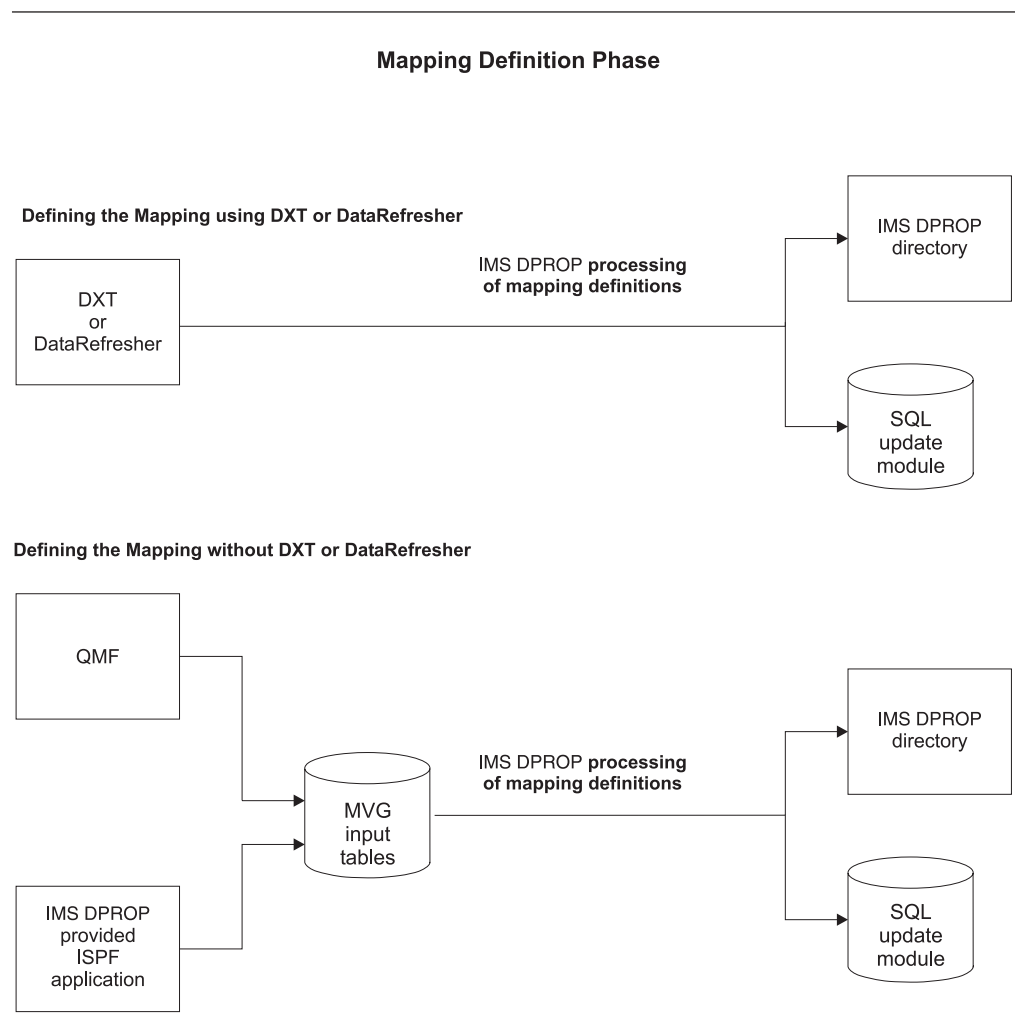


Figure 7. Mapping definition phase

Your request specifying what is to be mapped is called a propagation request. IMS DPROP stores the mapping information in the IMS DPROP directory. The directory is a set of DB2 tables containing information about IMS segments, DB2 tables, fields and conversions, warning messages, and other related information needed by IMS DPROP. For performance reasons, IMS DPROP maintains copies of these tables as Virtual Lookaside Facility (VLF) objects.

You cannot store propagation requests for both synchronous propagation and asynchronous propagation in the same IMS DPROP directory. You must have a separate IMS DPROP directory for each.

If you are using mapping case 1, 2, or 3 for IMS-to-DB2 propagation, IMS DPROP creates an SQL update module for each propagation request. The SQL update module contains static SQL statements required to propagate data to DB2.

For DB2-to-IMS synchronous propagation, IMS DPROP dynamically generates the DL/I update calls required to propagate data to IMS and therefore does not create a DL/I update module.

You can use DataRefresher to provide mapping definitions for propagation requests even if you do not extract IMS data with DataRefresher and even if you are performing one-way IMS-to-DB2 propagation.

Setup phase

Setting up IMS DPROP for propagation involves:

- Making adjustments to your DB2 and IMS systems
- Creating:
 - Transmission specification files (where propagation data streams and MQSeries queue manager information is supplied)
 - Status files

For the details of these items, see the *IMS DPROP Administrators Guide for MQSeries Asynchronous Propagation*.

- Extracting and loading your existing data for the first time

Activities involved in adjusting your system include registering databases that are a source of propagation and modifying the database definition. You also make adjustments that are specifically required for synchronous or asynchronous propagation.

When you extract and load the data you provide a consistent starting point for propagation.

If you implement one-way IMS-to-IMS propagation, you extract the data from the IMS source and load the data to the IMS target copy. The IMS target database copy is subsequently updated as source IMS database changes are propagated.

If you implement one-way IMS-to-DB2 propagation, you extract the data from IMS and load DB2 tables. The DB2 data is the copy to which subsequent IMS changes are propagated.

If you implement one-way DB2-to-IMS synchronous propagation, you extract the data from DB2 and load the IMS databases. If performing two-way synchronous propagation, you must decide in which direction to extract and load data.

You might also extract and load data during the maintenance and control phase of propagation, to synchronize your systems.

Tasks you must complete during the setup of IMS DPROP are summarized in Chapter 1, “IMS DPROP Administrative Tasks” in the *Administrators Guide* that pertains to the type of propagation that you are performing.

This section summarizes differences between extracting and loading associated with IMS-to-IMS, IMS-to-DB2 and DB2-to-IMS.

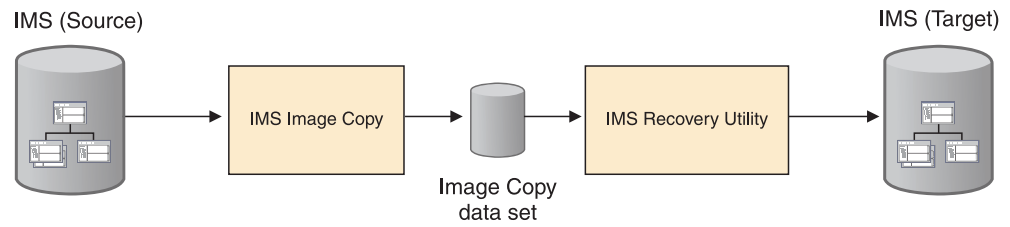
IMS-to-IMS extract and load

You can extract and load using your method of choice; for example:

- Using the Image Copy utility and IMS Recovery utility
- Using the HD Unload and HD Reload utilities

Figure 8 on page 17 shows the extract of IMS and load of IMS data using the examples above.

Method 1: Extract using Image Copy Utility



Method 2: Extract HD Unload Utility

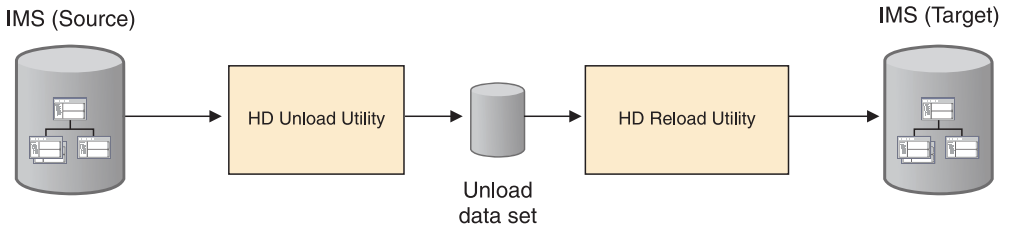


Figure 8. IMS-to-IMS extract using the Image Copy utility and HD Unload utility

Under normal circumstances, you probably maintain duplicate database copies of the IMS source and the IMS target. If this is the case, you must initially populate the IMS target copy prior to propagation, using your method of choice. The IMS source database must be quiesced in order to ensure integrity of the extracted source. If updates are made during the extract process, data inconsistencies can occur.

IMS-to-DB2 extract and load

You can extract and load using either:

- DataRefresher for the extract and the DB2 Load utility for the load
- Programs you provide

Figure 9 shows the extract of IMS data with DataRefresher and the load of the DB2 tables with the DB2 Load utility.

IMS-to-DB2 Extract and Load Phase: DataRefresher and the DB2 Load Utility

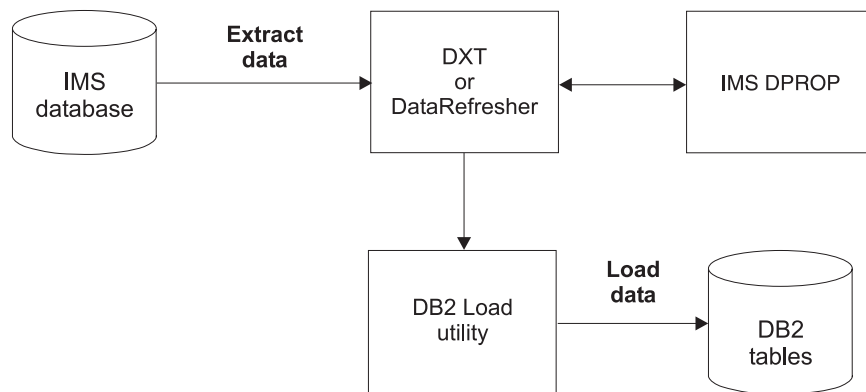


Figure 9. IMS-to-DB2 extract and load phase: DataRefresher and the DB2 Load utility

During the extract and load process, do not update the IMS database. Data must remain consistent between IMS and DB2. If DBRC share control is in effect, you can prevent unwanted updating by using the IMS DPROP Status Change utility (SCU). The SCU marks IMS databases as read-only. If you are performing the extract using DataRefresher, during extract IMS DPROP verifies that data is read-only, unless:

- You are performing user asynchronous propagation.
- Share control is not in effect.
- The IMS database is a data entry database (DEDB).

In these cases, you must devise your own method of ensuring data is not updated.

After loading the tables, run the DB2 COPY utility to establish a recovery point and backup copy of the DB2 data. Then run the DB2 RUNSTATS utility to optimize access to DB2 tables.

If you are performing asynchronous propagation and IMS and DB2 are on different systems, you can find additional considerations in the chapter titled “Extracting and Loading Data for IMS-to-DB2 Propagation” in the appropriate *Administrators Guide* for the type of propagation that you are performing. Refer to the following sections for additional considerations.

DB2-to-IMS extract and load

You can extract and load the data using either:

- IMS DPROP DL/I Load utilities (DLU)
- Programs you provide

Figure 10 shows DB2 data being extracted and IMS databases being loaded with the IMS DPROP DL/I Load utilities (DLU). You can use the SCU to mark the DB2 table spaces as read-only. The DLU verifies that data is read-only during the extract. See “DLU: Loading IMS databases from DB2 tables” on page 35 for more information on the DLU as it is used for loading data.

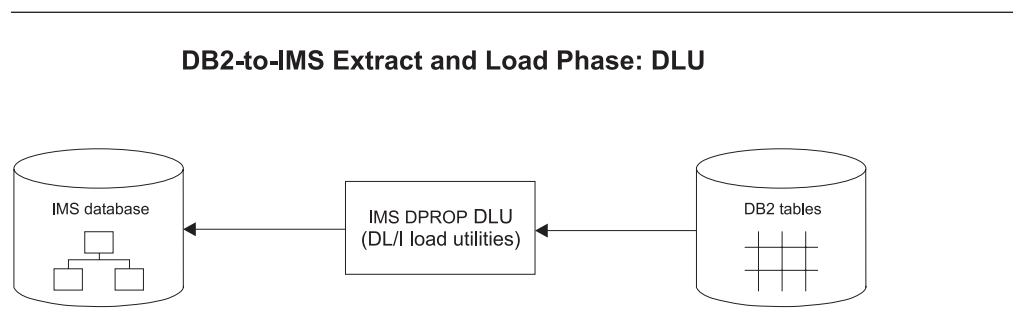


Figure 10. DB2-to-IMS extract and load phase: DLU

After loading the IMS databases, you might need to run IMS utilities. For example, if your IMS databases are involved in logical relationships, you must run the IMS Prefix Resolution and Prefix Update utilities. You should also create an image copy of the IMS databases to establish a recovery point and backup copy of the IMS data.

Propagation phase

After you extract and load data, you can begin data propagation. How you propagate depends on the propagation direction and mode you have set. For synchronous propagation, the system automatically begins propagation for you. For

asynchronous propagation, you complete a series of tasks. Tasks you must complete for asynchronous propagation during the propagation phase are summarized in Chapter 1, “Overview of Administrative Tasks” in the appropriate *Administrators Guide* for your propagation mode.

This section describes the events of propagation for:

- MQ-ASYNC propagation
- LOG-ASYNC propagation
- IMS-to-DB2 synchronous propagation
- DB2-to-IMS synchronous propagation
- User asynchronous propagation

IMS-to-IMS MQ-ASYNC propagation

Figure 11 shows how MQ-ASYNC propagation works when the IMS source and the IMS target are on different z/OS images.

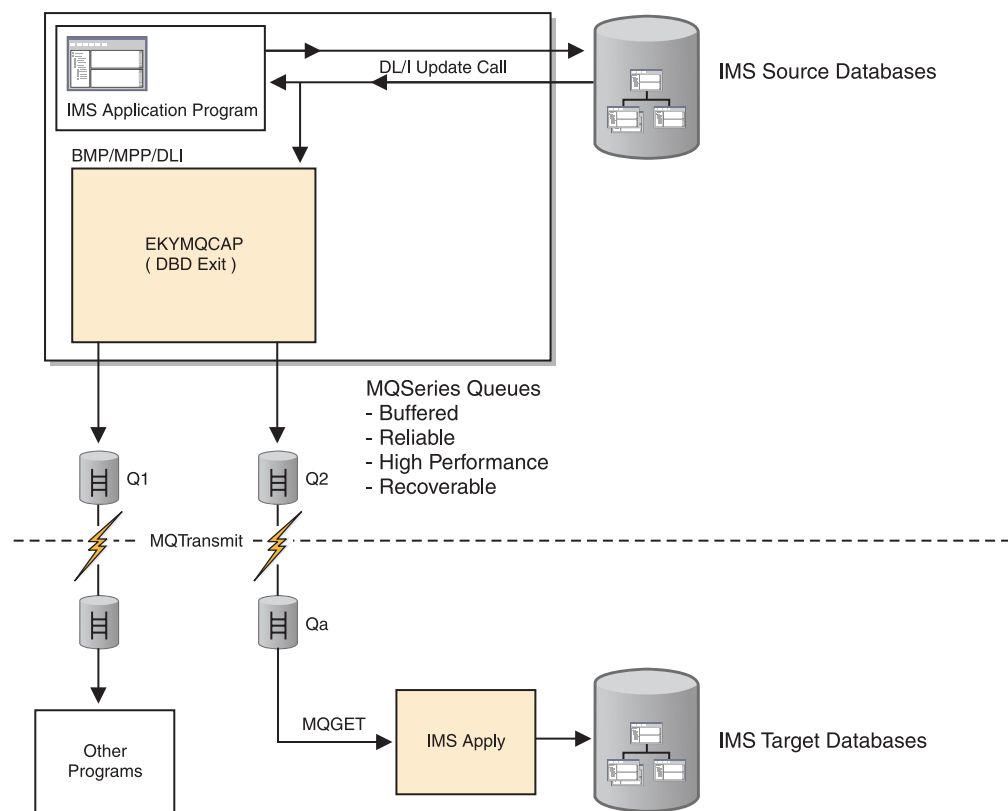


Figure 11. Propagation phase: MQ-ASYNC asynchronous IMS-to-IMS propagation

The following list describes Figure 11:

1. When an IMS application changes an IMS segment type to be propagated, DPROF captures the update using a Capture exit routine. This “capture” happens during the unit-of-work (UOW) of the IMS application. Control is passed to the Capture exit routine after the update to the segment is made.
2. The Capture exit routine builds an MQSeries message containing the information about the changed segment and places the message on an MQSeries message queue (labeled Q2 in Figure 11). The Capture exit routine can place the message on multiple queues (for example those labeled both Q1 and Q2), as shown in Figure 11.

3. If there were no problems during processing (updating the segment, building of the message, and placing the message on the MQSeries queue), the update to the segment is committed. The IMS application's UOW is now complete.

4. After the segment change is committed, MQSeries transmits the message to another MQSeries queue (labeled Qa in Figure 11 on page 19) on the system where the IMS Apply program is running. This second system is called the target system.

The changed IMS data flows from the Capture component to the IMS Apply program occurrences as part of one or more *propagation data streams* (PRSTREAMs). All changed data contained in one PRSTREAM is transmitted by means of the same MQSeries queues and is applied in First-In First-Out (FIFO) sequence by the same occurrence of the IMS Apply program.

5. The IMS DPROP IMS Apply program retrieves the message from queue Qa and updates the IMS target database.

With MQ-ASYNc IMS-to-IMS propagation, the IMS target database can reflect changes to the IMS source database rapidly. When conditions are optimal, the IMS target copy can reflect IMS source updates in near-real-time.

IMS-to-IMS LOG-ASYNc with MQ-ASYNc propagation

Figure 12 on page 21 shows how IMS-to-IMS propagation works when the LOG-ASYNc Selector is used to initially gather IMS source database changes. The source changes gathered by the Selector are written to MQSeries queues where they are read by the IMS Apply program and used to update the IMS target database. In the case shown, both the IMS source and target databases are on the same z/OS image.

One-Way IMS-to-IMS Propagation

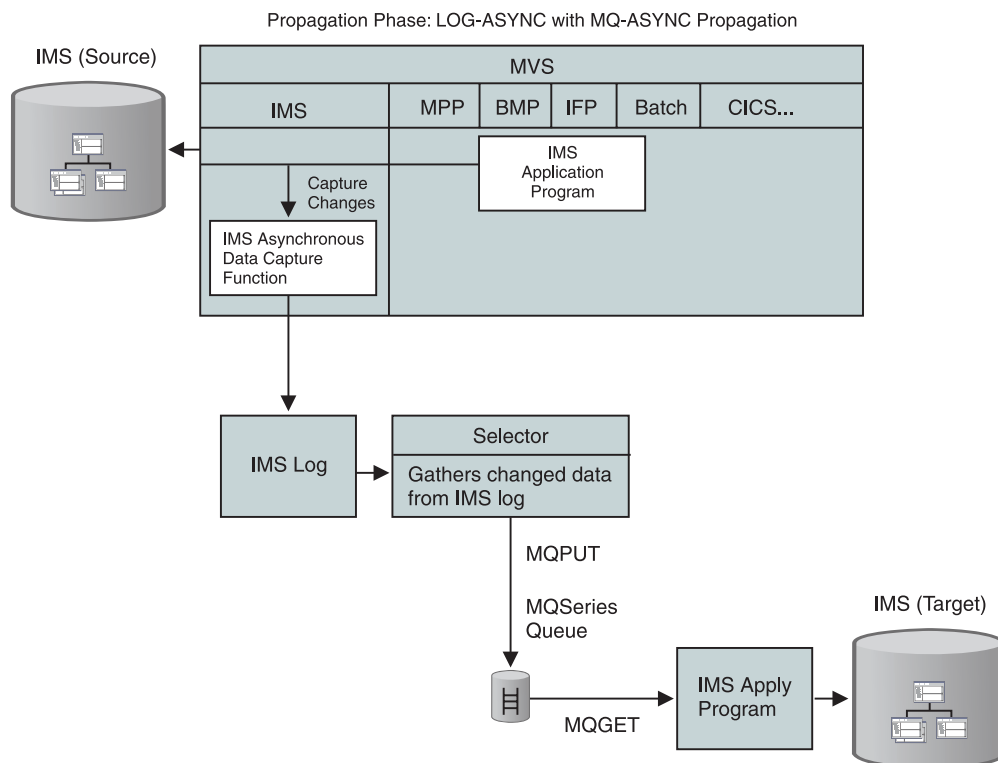


Figure 12. Propagation phase: LOG-ASYNC Selector used with MQ-ASYNC propagation

The following list describes Figure 12:

1. IMS application programs update IMS source databases. The IMS Asynchronous Changed Data Capture function writes updates to specified databases to the IMS log. Optionally, the modified IMS archive job filters out propagation log records and writes them to CDCDS.
2. When the Selector runs, it gathers and selects the changed data from the IMS log data sets or the CDCDS. The Selector calls the IMS DPROP Capture exit routine which builds an MQSeries message containing the information about the changed segment and places the message on an MQSeries message queue. The Capture exit routine can place the message on multiple queues.
3. If there were no problems during processing (building the message and placing it on the MQSeries queue), the update to each processed segment is committed. The changed IMS data flows from the Capture component to the IMS Apply program occurrences as part of one or more *propagation data streams* (PRSTREAMs). All changed data contained in one PRSTREAM is transmitted by means of the same MQSeries queues and is applied in First-In First-Out (FIFO) sequence by the same occurrence of the IMS Apply program.
4. The IMS DPROP Apply program retrieves the message from the queue and updates the IMS target database.

IMS-to-DB2 MQ-ASYNC propagation

Figure 13 on page 22 illustrates how MQ-ASYNC propagation works when IMS and DB2 are on different z/OS images.

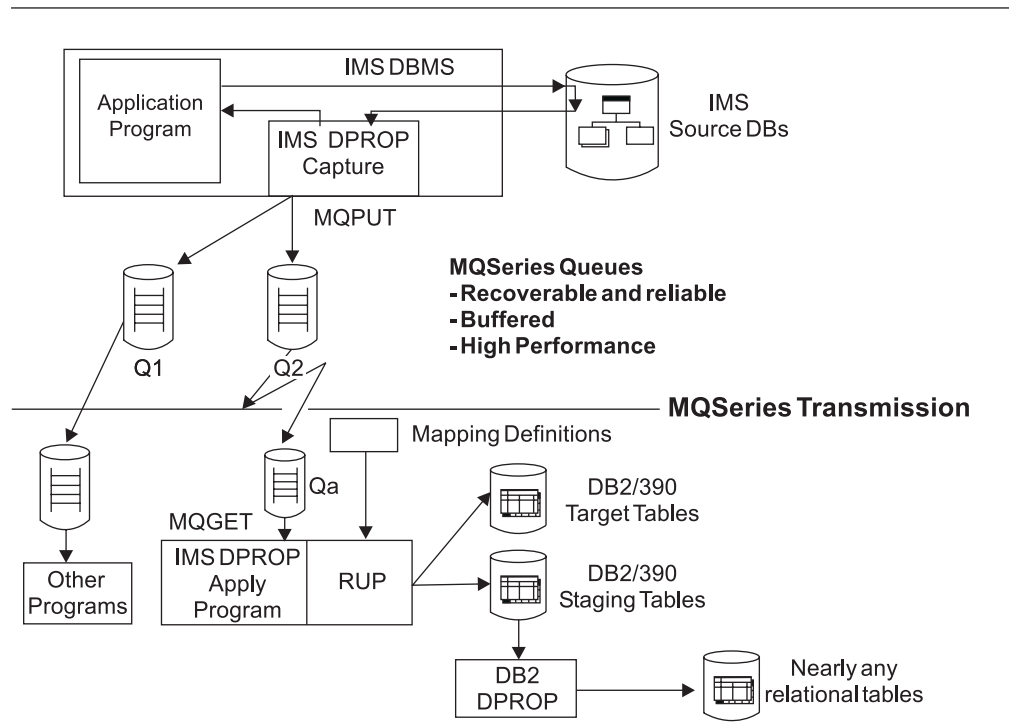


Figure 13. Propagation phase: MQ-ASYNC asynchronous propagation

The following list describes Figure 13:

1. When an IMS application changes an IMS segment type that is to be propagated, IMS DPROP captures the update using a Capture exit routine. This “capture” happens during the unit of work of the IMS application. Control is passed to the Capture exit routine after the update to the segment is made.
2. The Capture exit routine builds an MQSeries message containing the information about the changed segment and places this message on an MQSeries message queue (labeled Q2 in Figure 13). The Capture exit routine can place the message on multiple queues, as shown in Figure 13.
3. If there were no problems during processing (the update to the segment, the building of the message and the placing of the message on the MQSeries queue), the update to the segment is committed. At this point in time, the IMS application’s unit-of-work is complete.
4. After the segment change is finally committed, MQSeries transmits the message to another MQSeries queue (labeled Qa in Figure 13) on the system where DB2 is running. This second system is called the target system.
5. The IMS DPROP Apply program retrieves the message from queue Qa and passes it to the IMS DPROP Relational Update Program (RUP).
6. The RUP queries the mapping definitions that describe how the IMS change data “maps” to the target DB2 tables.
7. Using the information from the mapping definitions, the RUP updates the target DB2 tables.
8. If the target DB2 tables are staging tables, the DB2 DataPropagator product can propagate the change data to almost any other relational database.

With MQ-ASYNC propagation, the DB2 tables can reflect changes to the IMS database much faster than using LOG-ASYNC propagation. If conditions are optimal, the DB2 copy can reflect the IMS updates in near-real time.

IMS-to-DB2 LOG-ASYNC propagation

During the propagation phase in a LOG-ASYNC propagation environment, IMS saves application program updates in the IMS log and then IMS DPROP applies the saved changes to the DB2 tables. Figure 14 illustrates LOG-ASYNC propagation.

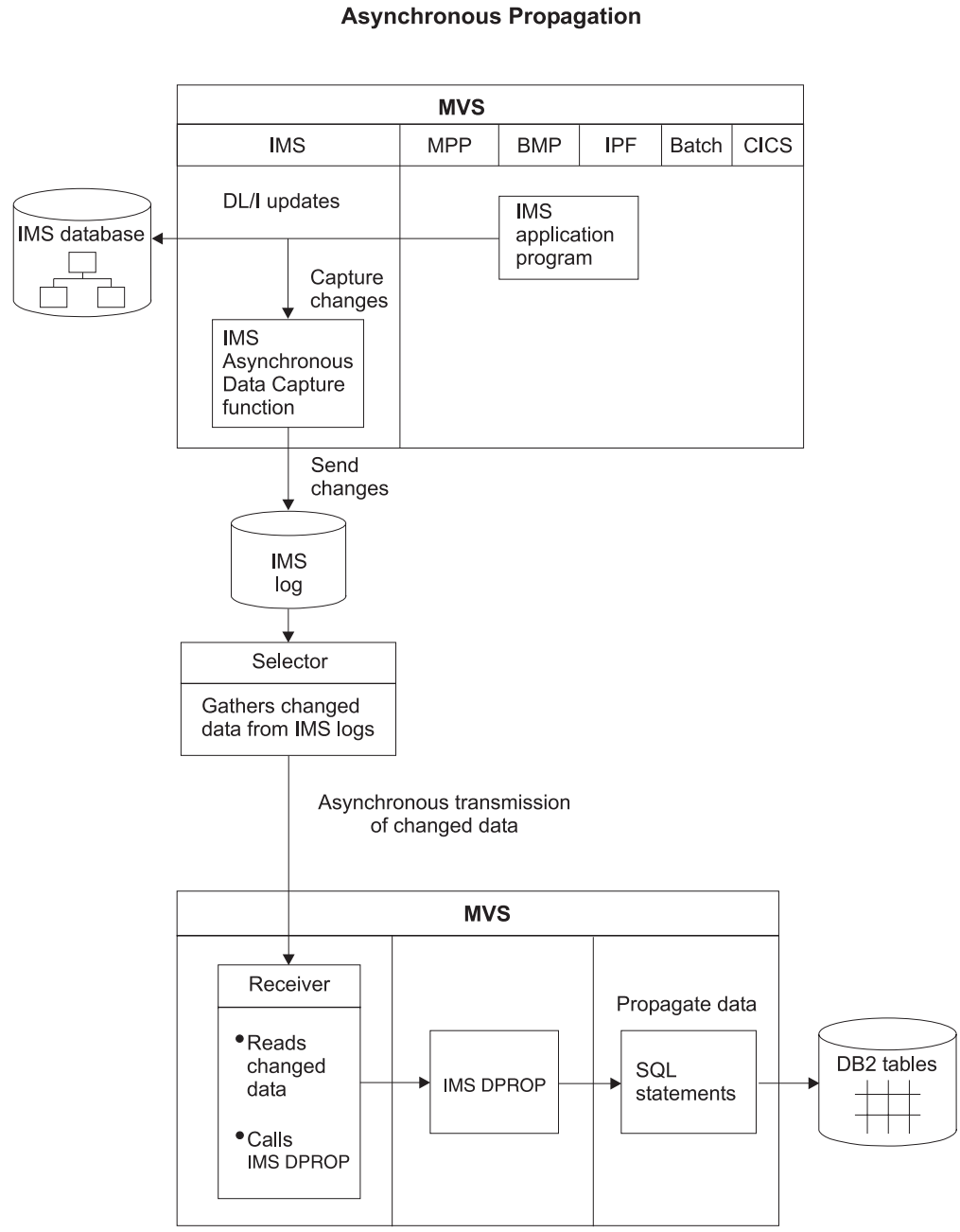


Figure 14. Propagation phase: LOG-ASYNC asynchronous propagation

IMS application programs update IMS databases. The IMS Asynchronous Changed Data Capture function writes updates to specified segment types to the IMS log. The modified IMS archive job filters out propagation log records and writes them to CDCDSs.

When the Selector runs, it gathers and selects the changed data from the IMS log data sets or the CDCDSs. The Selector creates PRDSs from the selected log files. When the Receiver runs, it reads the changed data in the PRDSs. IMS DPROP checks in the directory or a VLF copy of its directory to determine whether or not to do propagation and how to do it. If you use:

- Generalized mapping cases, IMS DPROP does the propagation and might call your Segment and Field exit routines during propagation.
- Your own mapping case, propagation is done by a Propagation exit routine you write. Your exit routine does the mapping, conversions, and asynchronous propagation.

In either case, the DB2 tables are updated.

Synchronous IMS-to-DB2 propagation

With synchronous propagation, IMS and DB2 must run under the same MVS image. During the propagation phase in an IMS-to-DB2 synchronous propagation environment, IMS application programs update the IMS database and changes are propagated to DB2 tables.

Figure 15 illustrates synchronous data propagation. When an IMS segment subject to synchronous propagation is updated, the IMS Data Capture function calls IMS DPROP with the changed data. IMS DPROP checks in the directory or a VLF copy of its directory to determine whether or not to do propagation and how to do it. If you use:

- Generalized mapping cases, IMS DPROP does the propagation and might call your Segment and Field exit routines during propagation.
- Your own mapping case, the Propagation exit routine you write does the propagation. Your exit routine does the mapping, conversions and synchronous propagation.

In either case, the DB2 tables are updated.

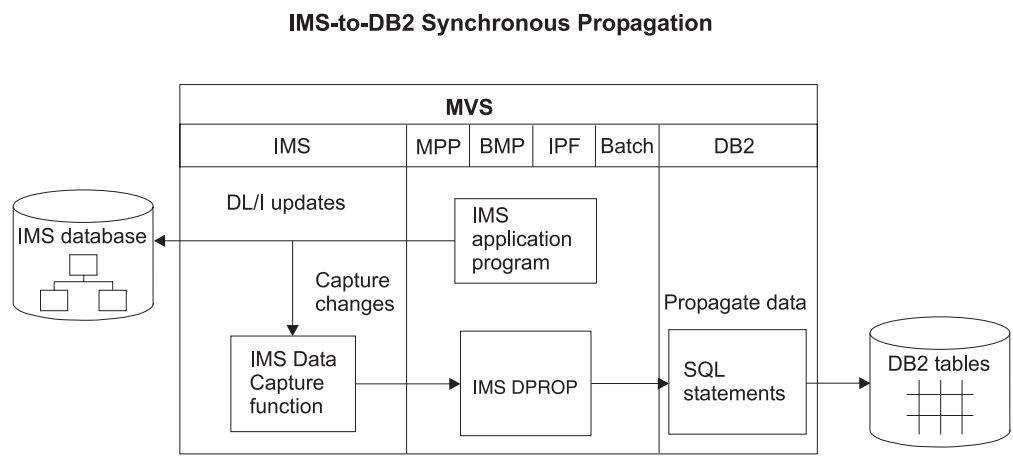


Figure 15. Propagation phase: IMS-to-DB2 synchronous propagation

Applications involved in synchronous propagation appear to DB2 and IMS as IMS/DB2 mixed mode programs. That is, applications appear to contain both IMS database calls and SQL statements. Standard connections, or threads, between IMS dependent regions and DB2 pass the SQL calls from IMS DPROP (or from your application) to DB2.

Synchronous DB2-to-IMS propagation

With synchronous propagation, IMS and DB2 must run under the same MVS image. During the propagation phase in a DB2-to-IMS synchronous propagation environment, application programs update the DB2 tables and changes are propagated to IMS databases. DB2-to-IMS synchronous propagation is supported only for SQL INSERT, UPDATE, or DELETE statements issued from an IMS application through DB2's IMS attachment facility. Any other changes to DB2 data, including those made by DB2 utilities or SQL updates issued from non-IMS attachments, are not propagated to IMS. Nor is synchronous propagation supported for updates to remote DB2 tables.

Figure 16 illustrates DB2-to-IMS synchronous propagation. When a DB2 table defined with the DATA CAPTURE CHANGES option is updated, the DB2 Data Capture function notifies IMS that DB2 table changes are captured in the DB2 log; this notification invokes IMS DPROP. IMS DPROP gets the captured DB2 data from DB2. IMS DPROP checks in a VLF copy of its directory (or in the directory itself if the information is not found in VLF) to determine whether or not to do propagation and how to do it. If you use:

- Generalized mapping cases, IMS DPROP does the propagation and might call your Segment and Field exit routines during synchronous propagation.
- Your own mapping case, the Propagation exit routine you write does the propagation. Your exit routine does the mapping, conversions, and synchronous propagation.

In either case, the IMS databases are updated.

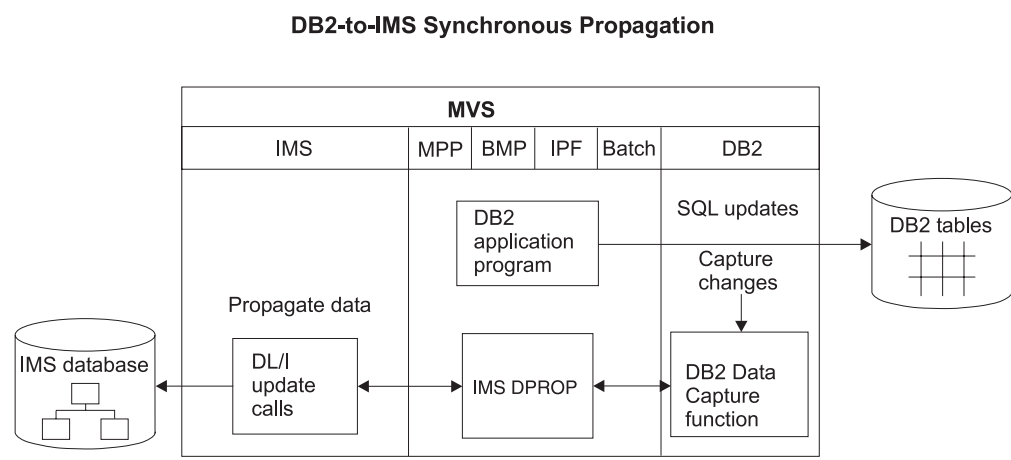


Figure 16. Propagation phase: DB2-to-IMS synchronous propagation

During the propagation phase in a synchronous environment, propagation can be:

- Suspended, for those few jobs whose performance is critical
- Deactivated, for problem situations
- Activated
- Turned off, so database repair jobs can execute without triggering synchronous propagation

If you suspend or deactivate propagation, you must resynchronize data before resuming synchronous propagation.

IMS-to-DB2 user asynchronous propagation

During the propagation phase in a user asynchronous environment, application program updates to the IMS database are saved. Then IMS DPROP applies the updates to the DB2 tables.

To implement user asynchronous propagation, you must write programs that complement IMS DPROP functions. You can use:

- The IMS Asynchronous Data Capture function
- A Data Capture exit routine
- Other alternatives

IMS asynchronous data capture function:

Notice

This section contains general-use programming interface and associated guidance information.

To use the Asynchronous Data Capture function, specify in the DBD that segment changes are to be written to the IMS log by the IMS Asynchronous Data Capture function. Figure 17 on page 27 illustrates the process.

IMS application programs update IMS databases. Updates to specified segment types are written by the IMS Asynchronous Data Capture function to the IMS log.

A program that you write (often called *the selector*) gathers and selects the changed data from the IMS log data sets. Another program that you write (often called *the receiver*) reads the changed data and passes it to IMS DPROP. IMS DPROP checks in the directory or a VLF copy of its directory to determine whether or not to do propagation and how to do it. If you use:

- Generalized mapping cases, IMS DPROP does the propagation and might call your Segment and Field exit routines during propagation.
- Your own mapping case, propagation is done by a Propagation exit routine you write. Your exit routine does the mapping, conversions, and user asynchronous propagation.

In either case, the DB2 tables are updated.

For additional information on user asynchronous propagation, see the *Customization Guide*.

User Asynchronous Propagation Using IMS Asynchronous Data Capture

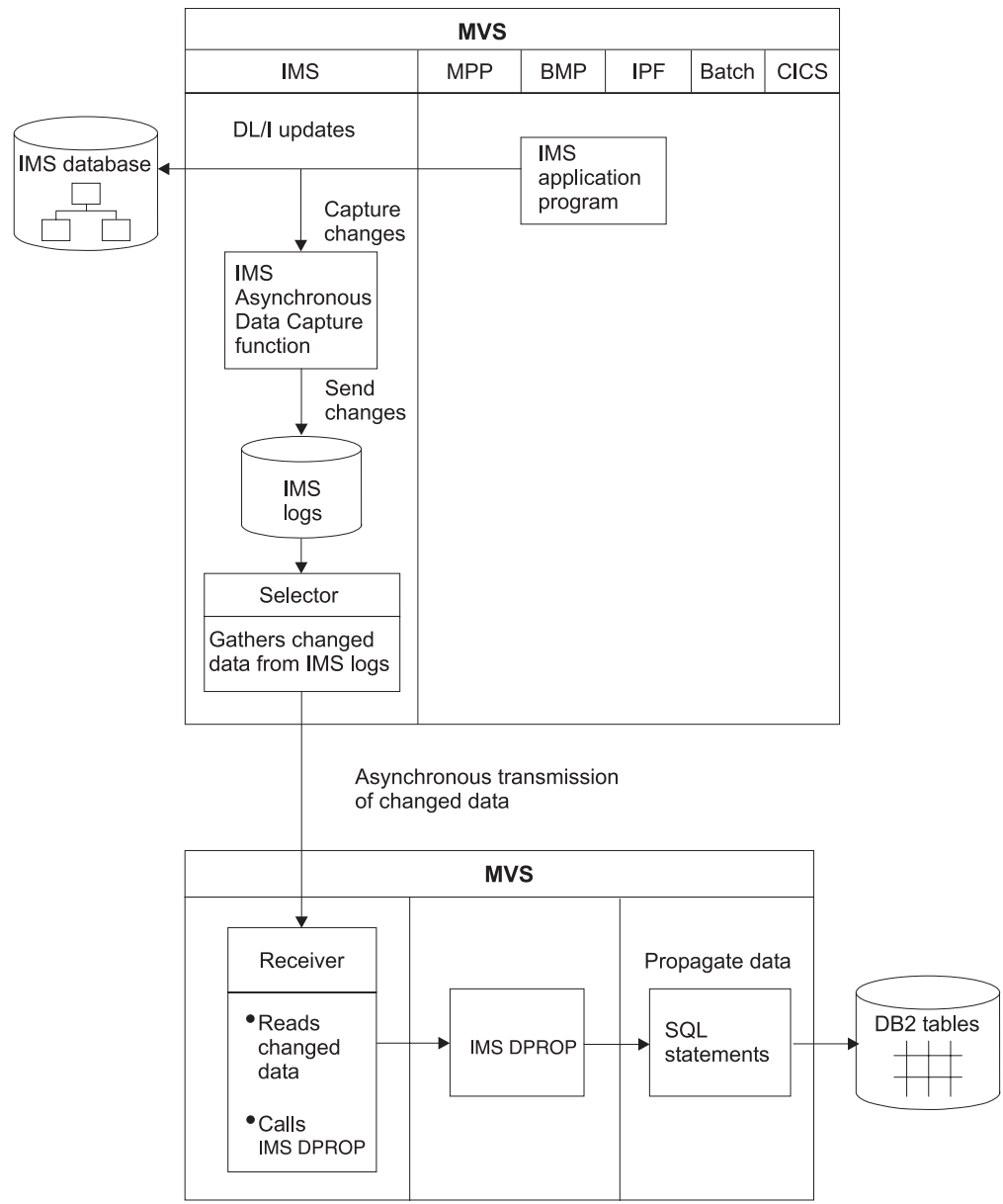


Figure 17. Propagation phase: User asynchronous propagation using IMS Asynchronous Data Capture function

Data Capture exit routine: To use the Data Capture exit routine, in DBD specify that segment changes are to be passed to a user-written Data Capture exit routine (sometimes referred to as the sender program). Figure 18 on page 29 illustrates the process.

IMS application programs update IMS databases. Updates to specified segment types cause the IMS Data Capture function to call your exit routine. The calls are part of the processing of the DL/I update calls in IMS application programs. Your exit routine must save the changed data. Save changed data using one of the following data objects:

- An IMS log

- The IMS full function database
- DEDB sequential dependent segments
- An MVS flat file

Another program you write (sometimes called the receiver program) reads the changed data and calls IMS DPROP. IMS DPROP checks in the directory or a VLF copy of its directory to determine whether or not to do propagation and how to do it. If you use:

- Generalized mapping cases, IMS DPROP does the propagation and might call your Segment and Field exit routines during propagation.
- Your own mapping case, propagation is done by a Propagation exit routine you write. Your exit routine does the mapping, conversions, and user asynchronous propagation.

In either case, the DB2 tables are updated.

User Asynchronous Propagation Using Your Data Capture Exit Routine

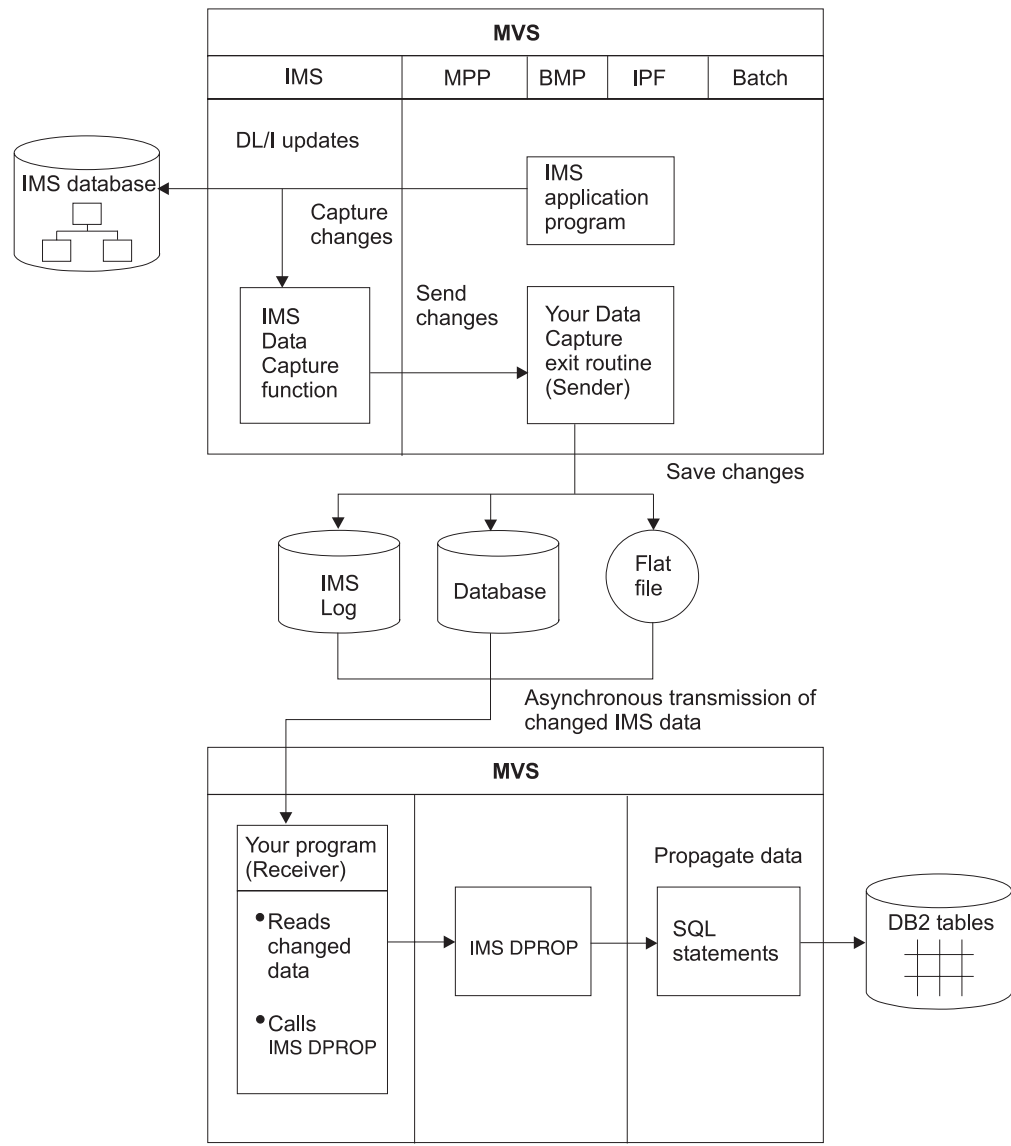


Figure 18. Propagation phase: User asynchronous propagation using your Data Capture exit routine

Other alternatives: You might decide to implement user asynchronous propagation differently from methods described in this section. Two alternative implementations are:

- If your sender program writes the changed data to the IMS log, you could use the Remote Recovery Data Facility (RRDF). RRDF has an IMS Log exit routine that gets control whenever records are written to the IMS log. RRDF transmits the log records containing the changed IMS data to the site where the DB2 tables reside. For more information on RRDF, refer to *Remote Recovery Data Facility Program Description and Operation*.
- If the IMS updates are made in an IMS online environment, the sender program can continuously send the changed IMS data to the receiver program by inserting the changed data into IMS output messages. If sender and receiver programs are on different MVS systems, you can send messages across MSC or ISC links.

For additional information on user asynchronous propagation, see the *Customization Guide*.

Maintenance and control phase

The maintenance and control phase occurs after you propagate data several times. Activities you perform during this phase are optional and performed only periodically. Activities include:

- Changing the status of propagation requests
- Checking the consistency of propagated data
- Cleaning up PRDSs and CDCDs
- Performing diagnosis

You can use a IMS DPROP utility, called the Consistency Check utility (CCU), to check the consistency of IMS and DB2 data. If mismatches are found, the CCU reports each inconsistency and creates repair statements for DB2 tables, IMS databases, or both.

IMS DPROP provides an audit trail and traces to help with problem determination.

Tasks you must complete during the maintenance and control phase are summarized in Chapter 1, “Overview of Administrative Tasks” in the appropriate *Administrators Guide* for the type of propagation that you are implementing.

IMS DPROP functions and components

IMS DPROP, using a combination of its components, performs the following functions:

- Creates mapping definitions
- Loads IMS databases from DB2 tables
- Propagates data from IMS-to-IMS
- Propagates data from IMS to DB2
- Propagates data from DB2 to IMS
- Changes the status of propagation requests
- Checks the consistency of propagated data
- Traces and audits IMS DPROP activities

This section summarizes IMS DPROP features, gives an overview of each function and describes the components used by each function.

IMS DPROP components

To propagate data using IMS DPROP, you must install one of the following components:

Synchronous

Required for synchronous propagation and user asynchronous propagation. This component must be installed on the same MVS image as the IMS databases and DB2 tables involved in the propagation.

Capture

Required for MQ-ASYNCR propagation. This component must be installed on the same MVS image as the IMS updating application.

Apply for IMS-to-IMS

Required for IMS-to-IMS MQ-ASYNCR propagation. This component must be installed on the same MVS image as the IMS target database.

Apply for IMS-to-DB2

Required for MQ-ASYNCR propagation. This component must be installed on the same MVS image as the DB2 target tables.

Selector

Required for LOG-ASYNCR propagation. This component must be installed on the same MVS image as the IMS DBRC.

Receiver

Required for LOG-ASYNCR propagation. This component must be installed on the same MVS image as the DB2 tables being propagated to.

Additional IMS DPROP components are:

Base Required for all types of propagation. This component must be installed either before or concurrently with all other IMS DPROP components and on the same systems as the components.

Utilities

Required by the Synchronous, Receiver, and Capture components. This component must be installed before or concurrently with the Synchronous, Receiver, and Capture components and on the same system as the Synchronous, Receiver, and Capture components.

MVG, MCE, and MVGU: Creating mapping definitions

Creating mapping definitions is part of the mapping and design phase described on page 13. Mapping definitions describe how mapping is done between IMS and DB2 data. The three components involved in generating and maintaining mapping definitions are:

- Mapping Verification and Generation (MVG)
- Map Capture exit (MCE)
- Mapping Verification and Generation utility (MVGU)

Mapping verification and generation

MVG validates the mapping information for each propagation request and stores the information in the IMS DPROP directory. You do not invoke MVG directly. The MVG is invoked internally by two IMS DPROP components: the MCE and the MVGU.

Map Capture exit

The MCE is a DataRefresher Map Capture exit routine provided by IMS DPROP. You call the MCE only if you provide mapping information through DataRefresher. DataRefresher calls MCE:

- During mapping definition, when it has an extract request that requires propagation. MCE validates the extract request, which is then called a propagation request.
- During extract of the IMS data, MCE checks that the IMS database from which data is to be extracted is read-only, and that no one is currently updating the IMS database.

Mapping Verification and Generation utility

The MVGU has three main functions:

- To create propagation requests if you do not use DataRefresher to provide mapping information. You must put the mapping information directly into a group of DB2 tables called the MVG input tables. To put the mapping information in the MVG input tables, you can use IMS DPROP ISPF panels, QMF, or an SQL application that you write. When the information is in the MVG input tables, you

invoke MVGU. MVGU validates the mapping information and then calls MVG, which creates the PR in the IMS DPROP directory.

- To delete or rebuild propagation requests in the IMS DPROP directory. Use the MVGU to delete or rebuild propagation requests, even if the mapping information was originally provided by DataRefresher.
- To revalidate existing propagation requests.

Capture exit routine

The IMS DPROP Capture exit routine (EKYMQCAP) is supplied with IMS DPROP and is used to capture changes to IMS databases as part of the MQ-ASYN type of propagation. EKYMQCAP runs in the IMS application's environment.

During DBDGEN, you specify the IMS Data Capture function to call the EKYMQCAP after changes are successfully applied to an IMS segment. IMS captures the changes and gives them to EKYMQCAP. EKYMQCAP is the IMS Data Capture exit routine provided by IMS DPROP.

As shown in Figure 19, the Capture exit routine is called by IMS every time an IMS application updates an IMS database or segment where EXIT=EKYMQCAP is specified.

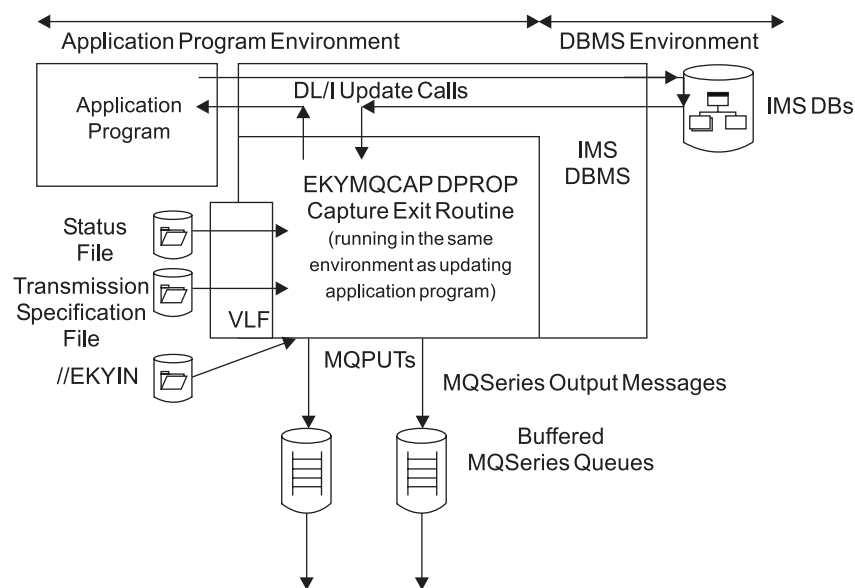


Figure 19. IMS DPROP's Capture Exit routine (EKYMQCAP)

The Capture exit routine:

1. Captures the update to the IMS database.
2. Checks the Status file (which you create) to see if propagation is turned on.
3. Checks the contents of the Transmission Specification file (which you create) for instructions on transmitting the update to the MQSeries queue.
4. Checks the IMS DPROP options file (//EKYIN) for additional option specifications.
5. Builds the MQSeries output message and places that message on the MQSeries queue.

All the steps described in the previous list take place within the IMS application's unit of work.

Apply program for IMS-to-IMS propagation

The IMS Apply program is supplied with IMS DPROP and is used to manage the process of applying IMS source data changes to IMS target data during MQ-ASYNC propagation as shown in Figure 20. The IMS Apply program runs on the same z/OS image as where the IMS target database resides.

The IMS Apply program:

1. Pulls the IMS DPROP MQSeries messages from the MQSeries queue. Each MQSeries queue is processed by one, and only one, Apply program.
2. Reads instructions from the two input data sets (//APPLYIN and //EKYIN)
3. Optionally, reacts to operator commands; for example, to stop.
4. Performs the updates to the IMS targets.

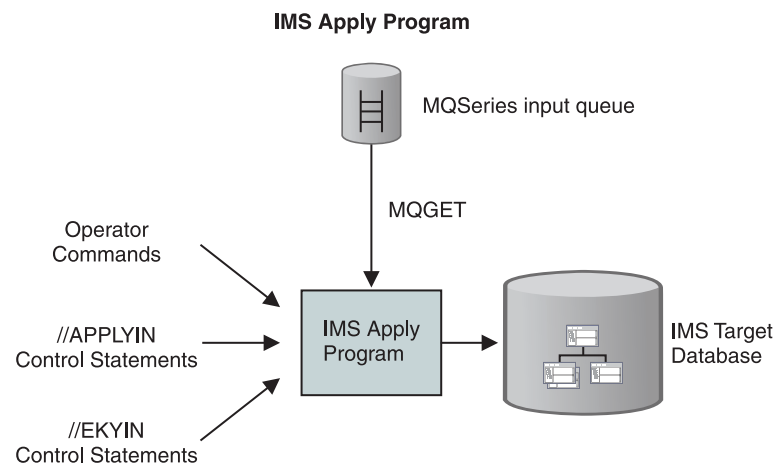


Figure 20. IMS DPROP Apply program

Apply program for IMS-to-DB2 propagation

The IMS DPROP Apply program for DB2 is supplied with IMS DPROP. and is used to manage the process of applying the IMS data changes to DB2, as part of the MQ-ASYNC type of propagation (see Figure 21 on page 34). The Apply program for DB2 runs on the same z/OS image as where the DB2 target tables reside.

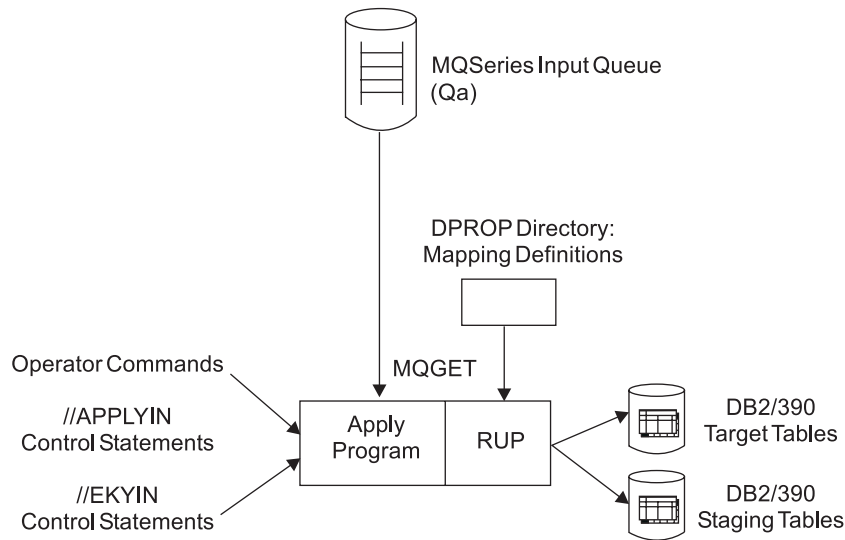


Figure 21. IMS DPROP's Apply program for DB2

As shown in Figure 21, the Apply program for DB2

1. Pulls the IMS DPROP MQSeries messages from the MQSeries queue (each MQSeries queue is processed by one and only one Apply program for DB2)
2. Reads instructions from two input data sets (//APPLYIN and //EKYIN)
3. Optionally, reacts to operator commands (to stop, for example)
4. Initiates the IMS DPROP Relational Update Program (RUP), which in turn implements the DB2 updates

RUP: Propagating data from IMS to DB2

IMS DPROP's relational update program (RUP) component does the actual IMS-to-DB2 propagation for:

- MQ-ASYNC propagation. See "Apply program for IMS-to-DB2 propagation" on page 33 for a description of how the Apply program for DB2 uses the RUP.
- LOG-ASYNC propagation. The changed data is written to the IMS logs and, if you set it up, to the CDCDSs. The Selector extracts the propagation log records from the log files and writes them to the Propagation Request data sets (PRDSs). The Receiver retrieves the changes from the PRDSs and passes them to the RUP.

The following set of utilities support the selection of propagation log records that relate to the PR mappings from IMS to DB2:

- Group Unload utility (GUU)
- SCF Compare utility
- SCF Apply utility
- PRDS Registration utility (PRU)
- Synchronous propagation. During DBDGEN you specify the IMS Data Capture function to call the RUP after changes are successfully applied to an IMS segment. IMS captures the changes and gives them to the RUP. The RUP is the IMS Data Capture exit routine provided by IMS DPROP.
- User asynchronous propagation. The changed data is either written to the IMS log or given to an IMS Data Capture exit routine that you write. Programs that you write read the changed IMS data, call the RUP and give the RUP the changed data.

Propagation involves either IMS DPROP's generalized mapping (mapping cases 1, 2, and 3 with the WHERE clause and the PATH data option) or mapping you provide through your Propagation exit routine. If propagation involves:

- Generalized mapping, the RUP does the propagation. Propagation using mapping cases 1, 2, and 3 may also involve calling Segment or Field exit routines that you write if you need to customize or extend existing mapping logic.
- A Propagation exit routine that you have written, the RUP provides various support functions, but your routine must do the actual mapping and propagation.

DLU: Loading IMS databases from DB2 tables

Loading IMS databases is part of the data extract and load phase. Use IMS DPROP DL/I Load utilities (DLU) to load IMS databases from DB2 tables. For DB2-to-IMS or two-way synchronous propagation, DLU can create or re-create IMS databases from the DB2 tables subject to synchronous propagation. DLU can also reestablish consistency if IMS and DB2 copies become inconsistent.

The DB2-to-IMS mapping done by DLU is based on the mapping definitions you previously created.

DLU loads segments propagated by IMS DPROP's generalized mapping. The DLU loads the segments from the tables to which data has been propagated. DLU provides limited support for segments propagated by user mapping and for segments that are not to be propagated. DLU can merge these other segments with the segments propagated by the generalized mapping. The merged segments are loaded into the IMS database.

Using the DLU during the extract and load process is described fully in the chapter titled "Extracting and Loading Data for DB2-to-IMS (DLU) Propagation" in the *IMS DPROP Administrators Guide for Synchronous Propagation*.

HUP: Propagating data from DB2 to IMS

DB2-to-IMS propagation is supported only in synchronous mode. IMS DPROP's hierarchical update program (HUP) does the actual DB2-to-IMS propagation. HUP is the DB2 Data Capture exit routine provided by IMS DPROP. You specify on a CREATE TABLE or ALTER TABLE statement that DB2 is to signal IMS whenever an SQL statement modifies a table subject to synchronous propagation. IMS then calls HUP, which passes the captured DB2 data to IMS.

Propagation involves either IMS DPROP's generalized mapping (mapping cases 1, 2, and 3 with the WHERE clause and PATH data option) or mapping you provide through your Propagation exit routine. If synchronous propagation involves:

- Generalized mapping, HUP does the propagation. Propagation using mapping cases 1, 2, and 3 may also involve calling Segment or Field exit routines that you write if you need to customize or extend existing mapping logic.
- A Propagation exit routine that you have written, the HUP provides various support functions, but your routine must do the actual mapping and propagation.

SCU: Changing the status of propagation requests

With synchronous propagation, you can change the status of propagation requests using the IMS DPROP Status Change utility (SCU).

A propagation request can be in one of these three states:

- Active (started)

- Inactive (controlled or emergency deactivation)
- Suspended

You might want to change a propagation request from active to suspended during lengthy batch jobs.

SCU also implements a variety of other service functions.

CCU, AUDU, and Trace: Monitoring propagation and determining problems

You can use the Consistency Checking utility (CCU), tracing, or the Audit Extract utility (AUDU) to monitor propagation and identify problems.

Checking the consistency of propagated data using the CCU

For mapping cases 1, 2, and 3, you can use the CCU, to check the consistency of propagated data. The CCU checks that the data exists and is consistent. If inconsistencies are found, the CCU:

- Generates a report describing the inconsistencies
- Creates files containing SQL repair statements and DL/I repair statements that you can use to fix inconsistencies

If you implement LOG-ASYNCR propagation, ensure that the IMS and DB2 data are at the same point in time. You might need to run the Selector and all relevant Receivers before running the CCU. If you implement two-way synchronous propagation, the CCU creates both SQL and DL/I repair statements. When the CCU detects an inconsistency, it does not know whether the IMS database or the DB2 tables are accurate.

For each inconsistency, you must determine whether or not:

- The IMS or DB2 data is accurate.
- To use the SQL repair statement, the DL/I repair statement, or your own procedures to restore consistency.

For more information about the CCU, refer to the chapter titled “Verifying Data Consistency (CCU)” in the appropriate *Administrators Guide* for the type of propagation you are performing.

Tracing

IMS DPROP provides a trace facility that can assist you in detecting errors. IMS DPROP traces:

- Changed IMS and DB2 data
- Propagating SQL and DL/I calls issued by IMS DPROP
- Calls to your exit routines
- IMS DPROP control blocks

You can activate tracing for the whole system using the SCU, or for a single job step using JCL. Trace information is written to the IMS log or to a trace data set.

You can also limit tracing of propagation activities from or to specific DBDs and segments by specifying a DEBUG= keyword on the type of information to be traced.

Refer to the *Reference* for complete information on IMS DPROP’s trace facilities.

Auditing using the AUDU

IMS DPROT records important propagation activities in an audit trail. IMS DPROT writes the audit trail to System Management Facility (SMF) records. If DB2 is installed, the AUDU can load these records into a DB2 table for diagnosis. You can then query these records using QMF or SQL applications.

The information in the audit trail is useful for problem determination. Refer to the *Reference* for complete information on IMS DPROT's audit facilities.

Comparison of audit and trace information

The audit trail table is a valuable centralized repository of historical information about propagation events. This information can be useful during problem determination by providing information about previous propagation-related events. Audit information differs, however, from trace information. Audit trail information is an overview of important propagation events. Trace information includes more detailed information.

Sysplex and data sharing

A Sysplex is a collection of MVS systems that can share data and DASD. IMS DPROT allows you to propagate data across a Sysplex, even when you have a mixture of IMS and DB2 systems. Within IMS DPROT, the Sysplex can be logically viewed as a single IMS or DB2 system. You can propagate data between IMS and DB2 applications.

How IMS DPROT works in a Sysplex

For synchronous propagation, a single IMS DPROT system must operate across multiple MVS systems, because any MVS system in the Sysplex can update IMS shared data. This means IMS DPROT control information must be shared data, existing on shared DASD.

The IMS DPROT control objects that must be shared are the:

- Status file
- Directory tables
- Global master timestamp (GMTS) PDS

For more information on IMS DPROT control objects, see the chapter titled "IMS DPROT Control Information and Environment" in the *IMS DPROT Administrators Guide for Synchronous Propagation*.

MQ-ASYN and LOG-ASYN propagation support Sysplex and data sharing because a single asynchronous IMS DPROT system image is not required to exist logically across multiple MVS systems.

However, the Selector must run on the same MVS system as the shared DBRC and the Receiver must run on the same MVS system as the shared DB2 in order to process work across all MVS systems in the Sysplex.

Within IMS DPROT, most of the Sysplex setup is done automatically during the installation process.

Planning for Sysplex

For more information on Sysplex systems and information on how to set up a Sysplex environment, refer to:

- *MVS/ESA Setting up a Sysplex*, GC28-1449

- *System/390 MVS Sysplex Overview*, GC28-1208
- *IMS/ESA Version 6 Release Planning Guide*, GC26-8744
- *DB2 for MVS/ESA Data Sharing: Planning and Administration*, SC26-3269

Environmental considerations

This section summarizes major IMS DPROP environmental considerations.

- IMS DPROP runs in an MVS environment.
- Data propagation is applicable for the following types of IMS databases:
 - HIDAM
 - HDAM
 - HISAM
 - SHISAM
 - DEDB

A DEDB sequential dependent segment (SDEP) can be propagated only by user mapping, not by the generalized mapping logic of IMS DPROP.

Propagation is *not* supported for HSAM, SHSAM, MSDB, GSAM.

Propagation is supported for secondary indexes; however, it is not supported for user data in secondary indexes when the secondary index is being treated as a separate database.

- Data propagation is not done during execution of:
 - IMS database recovery utilities
 - IMS HD and HISAM reload utilities
 - DB2 utilities
- Data propagation does not support all logical relationship delete rules. IMS DPROP rules are:
 1. Logical child segments must have a delete rule of VIRTUAL.
 2. Logical parent segments involved in IMS bidirectional relationships must have a delete rule of LOGICAL or PHYSICAL.
 3. Logical parent segments involved in IMS unidirectional relationships must have:
 - A delete rule of PHYSICAL (if propagated by TYPE=E propagation requests, which support one- and two-way propagation)
 - A delete rule of PHYSICAL or LOGICAL (if propagated by TYPE=L or TYPE=F propagation requests, which support only one-way IMS-to-DB2 propagation)

If you are currently using delete rules incompatible with data propagation, you must change them before performing data propagation. You might need to also change application programs.

- IMS DPROP generalized mapping logic requires that DB2 tables have a primary key.
- For one-way IMS-to-DB2 propagation, implementation of DB2 referential integrity relationships between propagated tables is optional.

The following sections describe environmental considerations for specific types of propagation:

- MQ-ASYNC
- LOG-ASYNC
- Synchronous
- User asynchronous
- IMS-to-IMS

MQ-ASYNC propagation environmental considerations

The following considerations apply to MQ-ASYNC propagation only:

- MQ-ASYNC propagation supports one-way IMS-to-IMS and IMS-to-DB2 propagation.
- MQ-ASYNC propagation requires two major components of IMS DPROP, Capture and Apply for IMS or Apply for DB2. It also requires, among other things, MQSeries and, for IMS Batch applications, RRS.

The Capture component captures IMS DB changes and writes them to MQSeries Queues. The Apply for DB2 component reads from MQSeries Queues and calls the RUP component of the IMS DPROP to update corresponding DB2 tables.

The IMS Apply component reads from MQSeries queues and updates the corresponding IMS target databases.

- The system in which Capture resides is called a source system and the system in which Apply resides is called a target system. The source system and the target system can either be in the same MVS image or in different MVS images.
- There are two operational modes for MQ-ASYNC propagation, near-real time propagation and point-in-time propagation. When the source system and the target system reside in different MVS images, the teleprocessing link and its bandwidth becomes an important consideration for the near-real time propagation scenario.
- If the target DB2 Tables are staging tables, further propagation to other relational databases in different platforms is possible using relational-to-relational propagation products, like IBM's DB2 DataPropagator.

LOG-ASYNC propagation environmental considerations

The following considerations apply to LOG-ASYNC propagation only:

- LOG-ASYNC propagation supports one-way IMS-to-DB2 propagation and, in addition, it supports IMS-to-IMS propagation by using the Selector as a front-end to feed MQ-ASYNC processing.
- The Receiver, which calls the RUP, can run in the following environments:
 - TSO foreground or TSO batch
 - MVS batch, using CAF Attach

You cannot issue DL/I calls from your programs, for example, from the three IMS DPROP exit routines.

- In case of an extended recovery facility (XRF) takeover for asynchronous propagation, you can run the Selector without any manual intervention because the primary IMS subsystem and the backup XRF subsystem use the same RECON data sets and log data sets.

You must have a IMS DPROP license for the alternate processor.

- You can propagate the same data to more than one DB2 system. Because one IMS DPROP is connected to only one DB2 system—the DB2 system that contains the IMS DPROP directory tables—you need one DPROP for each DB2 system that you want to propagate to.
- When IMS and DB2 are on different MVS images, the Receiver must reside on the same MVS image as DB2.

If the different MVS images are:

- At the same site (local MVS images), various data sets and libraries must either exist on shared DASD (so IMS DPROP can access it) or be copied to the MVS image where the Receiver resides.
- Not at the same site (remote MVS images), various data sets and libraries must be copied to the MVS image where the Receiver resides.

Synchronous propagation environmental considerations

The following considerations apply to synchronous propagation only:

- Programs performing synchronous IMS-to-DB2 propagation can run as message processing programs (MPPs), batch message processing (BMPs), IMS Fast Path (IFPs), or IMS batch programs.

Synchronous IMS-to-DB2 propagation does not run in the CICS environment. However, propagated DB2 tables and IMS databases can be accessed in read-only mode by CICS applications.

- Programs performing DB2-to-IMS synchronous propagation can run as MPPs, BMPs, IFPs, or IMS batch programs.

DB2-to-IMS synchronous propagation is supported only for SQL updates issued through DB2's IMS attachment facility. DB2-to-IMS synchronous propagation is not supported for SQL updates issued through DB2's TSO or CICS attachment facilities or the CAF. For example, updates done with QMF are not propagated.

If you need to propagate DB2 updates issued by programs using DB2's TSO attachment facility or the CAF, try running these programs with DB2's IMS attachment facility. You might need to protect propagated tables against DB2 updates implemented in TSO, CICS, or CAF environments; these environments are not supported for propagation. This is explained in "Protecting Propagated Tables from Nonpropagating SQL Updates" in *IMS DPROF Administrators Guide for Synchronous Propagation*.

DB2-to-IMS synchronous propagation is not supported for updates to remote DB2 tables. DB2-to-IMS synchronous propagation is not supported for updates to DB2 tables in a DB2 system other than the local DB2 system connected to the updating application program.

DB2-to-IMS synchronous propagation is also not supported for changes made to DB2 data by DB2 utilities.

- For one-way DB2-to-IMS and two-way synchronous propagation, implementation of DB2 referential integrity relationships between propagated tables is strongly recommended. The implemented DB2 referential integrity relationships should be compatible with the parent/child relationships between the IMS segments.
- In case of an XRF takeover for synchronous propagation, you must start DB2 manually on the alternate XRF processor.

You must have a IMS DPROF license for the alternate processor.

- IMS systems that update and the DB2 system that receives the update must run on the same MVS image. Therefore, if you are implementing intersystem data sharing, all IMS systems updating a propagated data base must run on the same MVS image.
- IMS DPROF does not support synchronous propagation of the *same* data to more than one DB2 system.
- More than one IMS DPROF system can run on one MVS image (for example, you can have one IMS DPROF system for testing and another for production).

You can invoke a single IMS DPROF system using more than one IMS system. All IMS systems accessing the same IMS DPROF system should use the same RECON.

Every IMS DPROF system propagates to one and only one DB2 system. Therefore, if you want to synchronously propagate *different* IMS data to DB2 tables in more than one DB2 system (for example, a production and an information center) you must define multiple IMS DPROFs, one for each DB2 system.

- IMS DPROP requires that all propagation implemented by one IMS dependent or batch region be implemented by the same IMS DPROP system to the same DB2 system. Synchronous propagation implemented by different IMS DPROP systems should be performed in a distinct set of IMS batch and dependent regions.
- IMS application programs must not issue ROLS calls with tokens or SETS calls. IMS batch programs must not issue any kind of ROLS calls. These restrictions also exist for IMS/DB2 mixed-mode programs.
- You can recover both your IMS and DB2 data copy to their most *recent* state. If you want to recover your IMS and DB2 data to a *previous* state, you must synchronize timestamp recoveries. There is no automated way to recover both IMS and DB2 to the same timestamp.
- If you are using DB2 with the bind package function, you can bind IMS DPROP programs as separate units called *packages*. Some IMS DPROP programs must be included in the propagating application's plan. When in the application's plan, you can make subsequent changes (for example, new or changed propagation requests) to the programs without having to re-bind the plans for all affected applications.

User asynchronous propagation environmental considerations

The following considerations apply to user asynchronous propagation only:

- Programs performing user asynchronous propagation can run as MPPs, BMPs, IFPs, or IMS batch programs. If you are using the IMS Asynchronous Data Capture function and capturing changed segments on the IMS log, you can also run programs performing IMS-to-DB2 propagation in a CICS DBCTL environment.
- User asynchronous propagation supports only one-way IMS-to-DB2 propagation.
- Your receiver program, which calls the RUP, can run in the following environments:
 - IMS batch or BMP region
 - TSO foreground or TSO batch using DB2's TSO attachment facility
 - DB2's CAF

In TSO and CAF environments, you cannot issue DL/I calls from your programs.

- You can propagate the same data to more than one DB2 system. Each IMS DPROP system is connected to only one DB2 system (the DB2 system that contains the IMS DPROP directory tables). If you want to propagate to more than one DB2 system, you must have a separate IMS DPROP system for each DB2 system.
- When IMS and DB2 are on different MVS images, IMS DPROP must reside on the same MVS image as DB2.
 - If the different MVS images are at the same site (local), the IMS DBDLIB for the IMS data being propagated must either exist on shared DASD (so IMS DPROP can access it) or be copied to the MVS image where IMS DPROP resides.
 - If the different MVS images are at different sites (remote), shared DASD for the IMS DBDLIB is not possible. Therefore, a copy of the IMS DBDLIB for the propagated data must be copied to the MVS image where IMS DPROP resides.

Synchronous propagation with asynchronous propagation environmental considerations

If you implement both synchronous propagation and asynchronous propagation, the following considerations apply:

- You can propagate the same data both synchronously and asynchronously from IMS to DB2.
- When you propagate data both synchronously and asynchronously, two IMS DPROPs are required.

IMS-to-IMS propagation environmental considerations

The following considerations apply to IMS-to-IMS MQ-ASYNCR propagation, only:

- IMS-to-IMS MQ-ASYNCR propagation requires the Capture and Apply components of IMS DPROP. It also requires MQSeries.
- The system in which Capture resides is called a source system and the system in which the IMS Apply resides is called the target system. The source system and the target system can either be in the same MVS image or in different MVS images.
- The Capture component captures source IMS DB changes and writes them to MQSeries queues. The IMS Apply component reads from MQSeries Queues and updates the corresponding target IMS DB.
- When the source system and the target system reside in different MVS images, the teleprocessing link and its bandwidth become an important consideration for a near real-time propagation scenario.

Chapter 2. Examples using IMS-to-IMS propagation

This chapter provides examples (in Problem/Solution format) of potential ways to perform IMS-to-IMS propagation.

Problem: Systems must be available 24 hours per day, everyday

An organization that provides banking services might have a requirement of this type. In such a case, ATM services must be continuously available; however, maintenance must be periodically performed on the IMS databases that support banking applications. For example, maintenance might be required for database expansion. In a different environment, the databases would be offline and unavailable while maintenance is performed. In a banking environment, this is not acceptable.

Solution: Keep systems available using IMS-to-IMS replication

The solution is implemented as follows:

1. Make an alternative copy of the primary IMS system and switch transactions to use this alternate system during the period of primary system maintenance.
2. While maintenance is performed on the primary system, capture updates to the alternate system using the changed data capture exit EKYMQCAP which is supplied by IMS DPROP MQ-ASYNCR. During this time, all IMS database changes are written as WMQ messages by the IMS DPROP Capture exit.
3. When maintenance of the primary system is complete, the changes made to the alternate system collected as WMQ messages are applied to the primary system by the IMS DPROP IMS Apply program. When all the updates from the alternate system are applied back to the primary system, switch transactions back to using the primary system once again.

Preparing the alternate system

An alternate IMS system is built by copying the primary IMS system.

The alternate system databases are defined as duplicates of the primary system but must be defined with the EXIT=EKYMQCAP specified on the DBD so that data changes to the alternate system can be captured. A full copy of the primary IMS system is performed using an efficient method such as FlashCopy® or PPRC technology.

As part of this copying methodology, an orderly transition between using the primary system and the alternate system must be made. During this transition the following actions are taken:

- Transactions on the primary system are stopped.
- The primary system is stopped.
- A full copy of the primary system database and associated IMS files are built as the alternate system.
- The alternate system is activated.

When the alternate system is fully established, IMS transactions are switched from the primary system and directed to the alternate system.

An overview of how the alternate system is prepared is shown below.

Overview of preparation of alternate IMS system

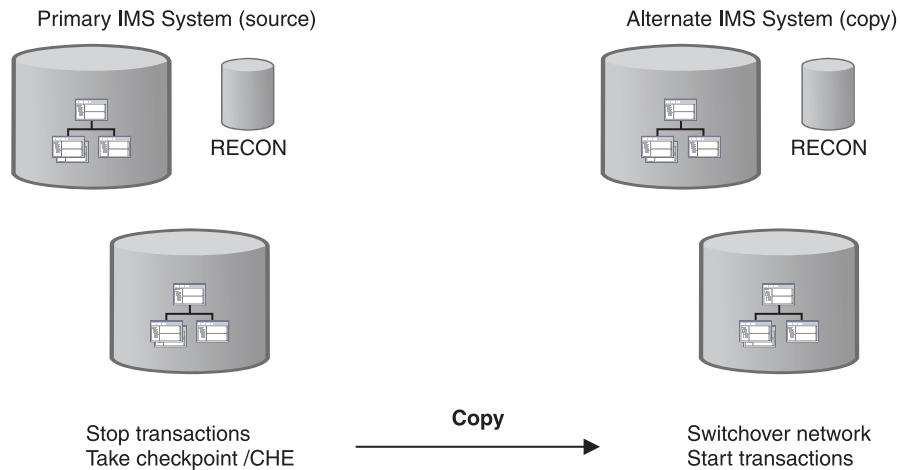


Figure 22. Preparation of alternate system for receiving IMS transactions

Providing service during the maintenance period

When updates are made to segments within the alternate system, the changed Data Capture exit provided by IMS DPROP (EKYMQCAP) is called. The Capture exit formats and writes WMQ messages containing details of the IMS updates. These WMQ messages containing application changes made to the alternate system databases are collected during the period of the primary system maintenance. The figure below shows service provided during the period of primary system maintenance.

Providing service during the Primary System maintenance period

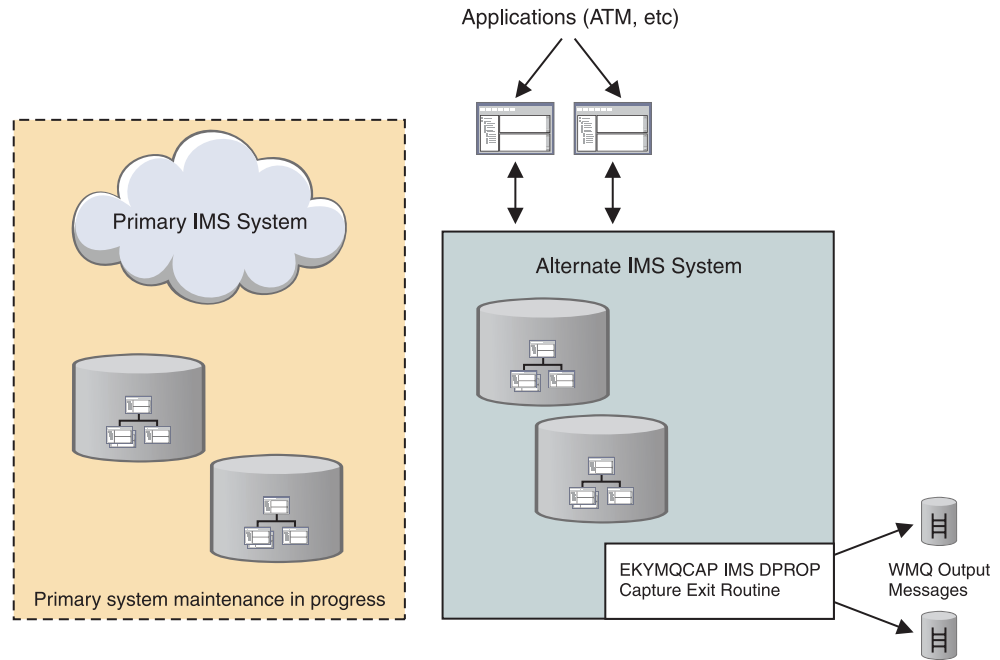


Figure 23. Service provided during primary system maintenance period

Switching back to the primary system from the alternate system

After the maintenance work is complete, restart the primary system. IMS database changes which were made on the alternate system during the maintenance period and were stored as WMQ messages, may now be used to update the primary system.

The IMS Apply program, supplied by IMS DPROP, is activated to process stored WMQ messages and update the primary system databases with updates that were made to the alternate system. When the IMS Apply program processes all the WMQ messages in the backlog and the primary system catches up with the changes made to the alternate system the switch back to the primary system may be made. Transactions are once again directed to the primary system, and the alternate system is retired.

The primary system maintenance is now complete. All IMS database update activity that was processed on the alternate system during the maintenance period has been applied to the primary system. Maintenance to the primary system has been completed with minimal impact to the applications, primary system users, and customers.

The process of switching back to the primary system after a period of outage for maintenance is shown in the following figure.

Overview of switching back to primary system from alternate system

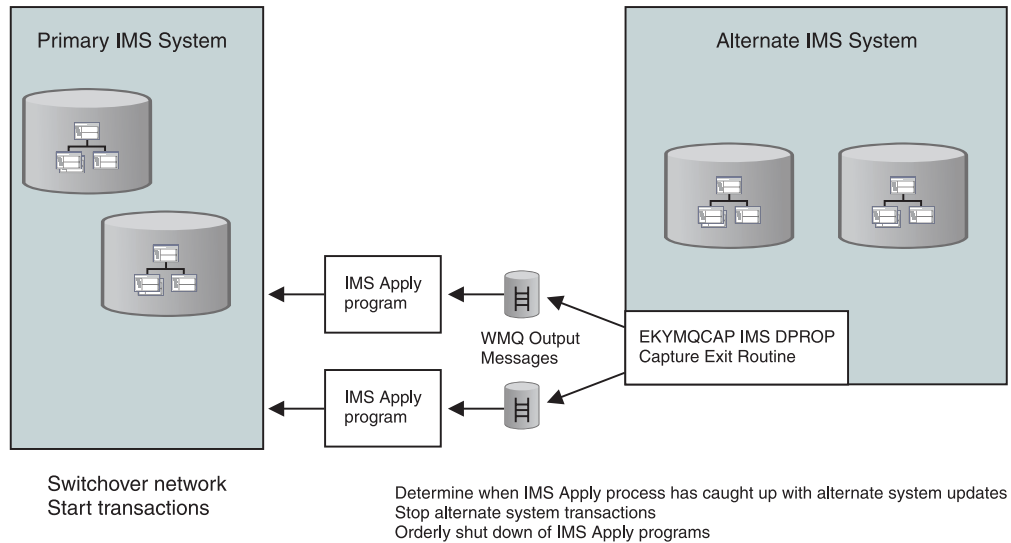


Figure 24. Switching back to primary system from alternate system

Problem: A shadow copy is required of some or all primary IMS system databases located at a remote site.

If an organization wants to enhance disaster recovery possibilities or represent the state of databases at their quiesce points, this can be accomplished by constructing shadow copies of the databases.

Solution: Use IMS-to-IMS replication features to construct a copy of the databases

Remote copies can be maintained so that data is propagated in near real-time or propagation can occur so that copies are up-to-date based on a chosen point-in-time.

The solution is implemented as follows:

1. Make an duplicate copy of the primary IMS system (which we will call the alternate system).
2. Capture updates to in one of the following ways:
 - Use the changed data Capture exit (EKYMOCAP) supplied by IMS DPROP MQ-ASYNC to write changes as WMQ messages.
 - Capture IMS data changes to the IMS log utilizing the IMS Log exit and then write the captured changes to the IMS log as WMQ messages.
3. Propagate the captured changes to the alternate system. As noted, IMS database changes will be written as WMQ messages by the IMS DPROP Capture exit. The IMS DPROP Apply program will interpret the WMQ messages and update the alternate system databases to reflect changes to the primary system.

Note: Utilizing the IMS DPROP Log-ASYNCR Selector feature for initial sourcing of IMS changed data allows IMS-to-IMS functionality to be provided when the source updates do not originate from an IMS updating environment. For example, the updates might be from CICS/DBCTL rather than from IMS TM transactions.

Preparing for propagation to the alternate target system

An alternate IMS system is built by copying the primary IMS™ system.

The alternate system databases are defined as duplicates of the primary source system. As part of this copying methodology, an orderly procedure must be used to allow data integrity between the two database copies which are to be maintained. At a certain point, the primary IMS source system is stopped, and a copy of the source system is made.

Before the primary system is once again made available, the following features must be enabled:

- The IMS DPROP data Capture exit (EKYMQCAP) must be enabled for any of the DBDs of the primary IMS system for databases that are to be propagated.
- If the IMS logs are to be used as the initial source for captured IMS changed data, then the IMS Log exit must be enabled for any of the DBDs of the primary IMS system for databases that are to be propagated.

Subsequently, the primary IMS system may be restarted, and ongoing IMS data changes can be captured for propagation purposes.

Propagating IMS data to the alternate target system

During preparation for propagation to the alternate target system, the decision was made to use either the direct IMS DPROP data Capture exit (EKYMQCAP) or the IMS Log exit as initial source for IMS changed data.

Using the IMS DPROP EKYMQCAP exit for primary data capture: When using the data Capture exit, EKYMQCAP, the following tasks are completed:

1. IMS data changes are written as WMQ messages.
2. The IMS Apply program, supplied by IMS DPROP, reads the WMQ messages written by the EKYMQCAP exit and applies the changes from the primary (source) system databases to the alternate (target) databases.

When using this capture and apply process, continuous near real-time propagation from IMS source to IMS target system may be implemented. Alternatively, through features supplied with IMS DPROP, a point-in-time replication may also be chosen.

This method of using the EKYMQCAP exit with the IMS Apply program for IMS-to-IMS propagation is shown in the figure below.

Overview of ongoing Propagation to alternate IMS target system

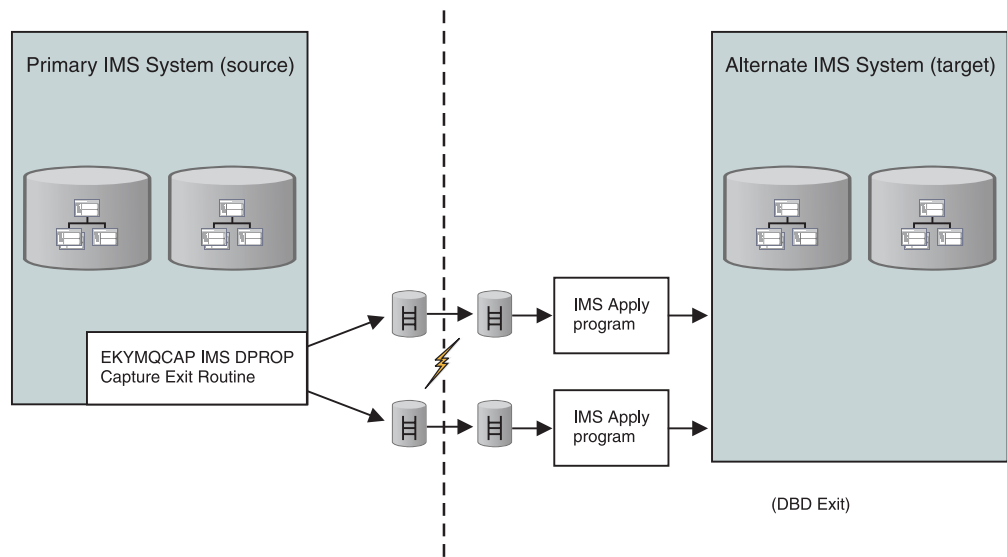


Figure 25. Overview of process using Capture and Apply exits for outgoing propagation

Using the IMS Log exit for primary data capture: When using the Log exit for capturing primary (source) data, the following tasks are completed:

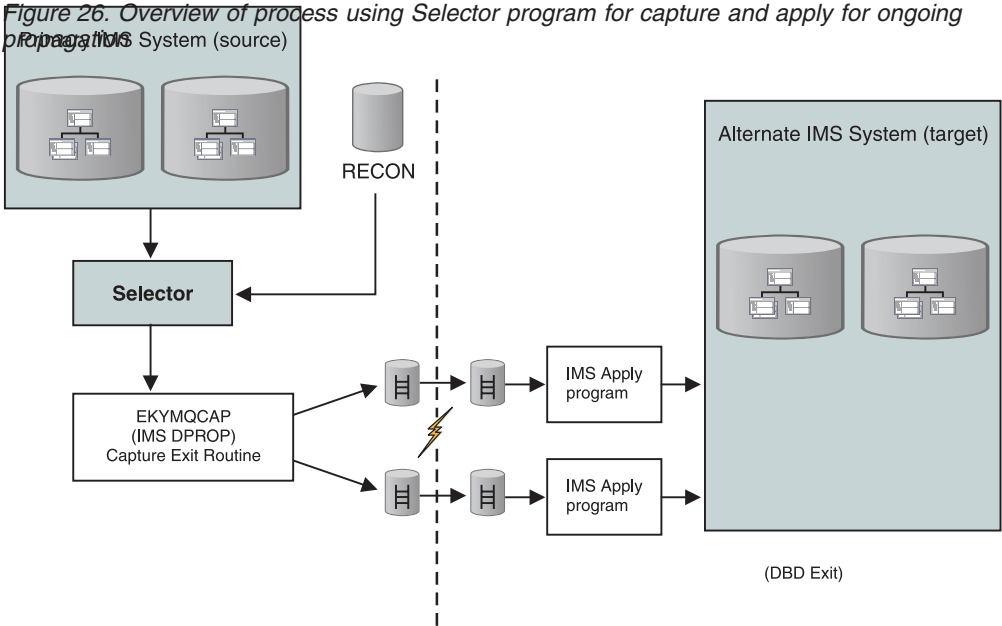
1. IMS data changes are captured as special changed data records and written to the IMS logs. The IMS DPROP Log-ASYNCR Selector program processes the changed data capture log records and calls the IMS DPROP EKYMQCAP exit to write WMQ messages containing IMS source database changes.
2. The IMS Apply program, supplied by IMS DPROP, reads the WMQ messages written by the EKYMQCAP exit and applies the changes from the primary (source) system databases to the alternate (target) databases.

When using this hybrid capture and apply process, which uses features of both IMS DPROP Log-ASYNCR and IMS DPROP MQ-ASYNCR propagation, near real-time updates between source and target will not be available. The rate at which propagation occurs between source and target databases will be driven by when IMS archived batch logs or when the batch logs become available to the IMS DPROP Selector program. Through features supplied with IMS DPROP, a point-in-time replication can be chosen with this propagation technique.

The hybrid propagation method described in this section is shown in the figure below.

Overview of IMS-to-IMS Propagation utilizing the Selector during the Capture process

Figure 26. Overview of process using Selector program for capture and apply for ongoing propagation



Chapter 3. Typical uses of IMS DPROP with IMS-to-DB2 propagation

IMS DPROP is useful if you need to access the same data using both the IMS DL/I call interface for IMS and the DB2 SQL interface. Propagation with IMS DPROP supports:

- Coexistence of DL/I operational applications and SQL applications
- Incremental migration of your current DL/I applications to SQL

Coexistence of IMS operational applications and DB2 applications

IMS DPROP allows you to use IMS operational applications with DB2 operational applications in a variety of combinations. You can use:

- DL/I update access and SQL read-only access
- DL/I read-only access and SQL update access
- DL/I update access and SQL update access

Coexistence of DL/I update access and SQL read-only access

You can use either one-way synchronous propagation, MQ-ASYNCR propagation, LOG-ASYNCR propagation, or user asynchronous propagation, if your applications need DL/I update access or SQL read-only access to the propagated data.

When operational DB2 applications need up-to-date data, IMS DPROP users can:

- Implement one-way synchronous propagation if the commits of data on IMS and DB2 have to be in the same unit-of-work.
- implement MQ-ASYNCR propagation using near-real time propagation mode if the commits of data in IMS and DB2 do not have to be in the same unit-of-work and a few seconds delay between the commit on IMS and separate unit-of-work commit on DB2 is acceptable.

Otherwise, IMS DPROP users can implement:

- IMS Synchronous Propagation
- MQ-ASYNCR propagation in either near-real time mode or point-in-time mode.
- LOG-ASYNCR propagation

To understand the advantages and disadvantages of synchronous and asynchronous IMS-to-DB2 propagation, refer to Chapter 4, "Trade-offs between synchronous and asynchronous with IMS-to-DB2 propagation," on page 55.

Coexistence of DL/I read-only access and SQL update access

Use one-way DB2-to-IMS synchronous propagation, if your applications need both DL/I read-only access and SQL update access to propagated data. If you currently have IMS applications updating the propagated data, you must either:

- Convert them to update the propagated data through the SQL interface
- Replace them with new DB2 applications

For each group of logically related IMS databases, you must simultaneously:

- Convert or replace the updating IMS applications
- Implement one-way DB2-to-IMS propagation

If this is not possible, consider using one of the following two-way propagation methods:

- Start implementing one-way DB2-to-IMS synchronous propagation and simultaneously convert or replace your updating IMS applications.
- Implement one-way IMS-to-DB2 synchronous propagation first, then switch to one-way DB2-to-IMS synchronous propagation later.

This method allows you to gain experience with IMS DPROP and (one-way IMS-to-DB2) synchronous propagation before converting or replacing your updating IMS applications and implementing one-way DB2-to-IMS synchronous propagation. This method might reduce your risk.

Coexistence of DL/I update access and SQL update access

Use two-way synchronous propagation, if your applications need both DL/I update access and SQL update access to propagated data.

You do not need to convert or replace your updating IMS applications with this approach. However, there are disadvantages. If the two copies of the propagated data become inconsistent, it is difficult to resynchronize the data, because there is no master. The IMS DPROP CCU describes the inconsistencies in detail and generates both DB2 and IMS repair statements you can use to restore consistency. However, IMS DPROP does not know if the IMS or DB2 copy is accurate. You must determine:

- Which copy is accurate for each inconsistency
- If and how the two data copies can be resynchronized in your specific environment

You can implement two-way synchronous propagation in a no-master-copy environment. Or you can implement one-way IMS-to-DB2 synchronous propagation first and switch to two-way synchronous propagation later.

Implementing incrementally allows you to gain experience with IMS DPROP and (one-way IMS-to-DB2) synchronous propagation before switching to two-way synchronous propagation.

If your environment does not require a master copy of the propagated data but you prefer to have a designated master:

1. Implement two-way synchronous propagation.
2. Convert or replace your updating IMS applications to updating DB2 applications, as appropriate (for example, one program or one group of segments at a time).
3. When all IMS applications are converted, switch from two-way to one-way DB2-to-IMS synchronous propagation.

Incremental migration of IMS applications

If you want to carry out a gradual, phased migration of your existing IMS applications to DB2, you can implement either of the following types of synchronous propagation:

- IMS to DB2
- DB2 to IMS
- Two-way

Or, you can implement one type of propagation and then switch to another. For example, you can:

- Implement one-way IMS-to-DB2 synchronous propagation and then switch to one-way DB2-to-IMS synchronous propagation

While doing one-way IMS-to-DB2 synchronous propagation, you can:

- Migrate your read-only IMS applications in stages
- Implement new read-only DB2 applications

When you switch from one-way IMS-to-DB2 to one-way DB2-to-IMS synchronous propagation you must simultaneously convert or replace all IMS applications that update a specific set of logically-related IMS databases. You can do this switch at a different time for each set of logically-related IMS databases.

- Implement one-way IMS-to-DB2 synchronous propagation, then switch to two-way synchronous propagation, and finally switch to one-way DB2-to-IMS synchronous propagation:
 1. Switching from one-way IMS-to-DB2 synchronous propagation to two-way synchronous propagation allows you to migrate your updating IMS applications in stages. However, as described in “Coexistence of DL/I update access and SQL update access” on page 52, you must determine if and how your installation can resynchronize the two data copies if they become inconsistent.
 2. After you have converted or replaced your last updating IMS application, you can switch from two-way to one-way DB2-to-IMS synchronous propagation.

Chapter 4. Trade-offs between synchronous and asynchronous with IMS-to-DB2 propagation

When deciding whether to implement asynchronous propagation or synchronous propagation, consider the advantages and disadvantages of each.

You can define a IMS DPROP system for MQ-ASYNC propagation, LOG-ASYNC propagation, synchronous propagation or user asynchronous propagation. However, if you want to use more than one type of propagation, you must create a IMS DPROP system and directory for each.

Synchronous propagation supports both one-way and two-way propagation and is usually used if you use the DB2 copy of the data for new operational DB2 application programs.

MQ-ASYNC propagation supports one-way IMS-to-DB2 propagation and is usually used if you use the DB2 copy of the data for decision-support applications that need updates in near-real time.

LOG-ASYNC and user asynchronous propagation supports one-way IMS-to-DB2 propagation and is usually used if you use the DB2 copy of the data for decision-support applications that do not need updates in real time.

Advantages and disadvantages of MQ-ASYNC propagation

MQ-ASYNC propagation supports only one-way IMS-to-DB2 propagation.

The first part of the MQ-ASYNC propagation process (the capturing of the changed data) is done within the same unit-of-work (UOW) of the IMS application program that is performing the update. The second part of MQ-ASYNC propagation process, the propagation of changed data to DB2, is performed outside the commit scope, or UOW, of the IMS update.

MQ-ASYNC propagation can occur in the same or a different MVS image, or in the same or different central processor complexes (CPCs), local or remote.

Two modes of operation for MQ-ASYNC propagation are available, near-real time propagation or point-in-time propagation.

Advantages

The advantages of MQ-ASYNC propagation include:

- You can develop applications that use SQL to read the DB2 copy of the propagated IMS data.
- IMS and DB2 can reside on different MVS images.
- You can operate your DB2 system independently of IMS. If problems develop on the DB2 side, you don't have to stop IMS: IMS is insulated from DB2 problem.
- IMS, MQSeries, and DB2 operate asynchronously to each other. Each can operate at its optimal performance without being delayed by others. The performance impact on propagating applications is minimized.
- You can choose to apply DB2 updates during periods of low machine activity.
- With near-real time mode of propagation, the DB2 data is only a few seconds behind the current IMS data (assuming the TP links performance support the transportation of data adequately when the target system and source system reside at different MVS images).

Disadvantages

The disadvantages of MQ-ASYNC propagation include:

- The DB2 version of the propagated data might not reflect the latest version of the data. However, this is much less of a problem for MQ-ASYNC propagation than for LOG-ASYNC, especially when the near-real time mode of propagation is used.
- Neither one-way DB2-to-IMS nor two way propagation is supported.
- The scope of the commit for the DB2 database updates can not be guaranteed to be exactly the same as the scope of the commit for the IMS DB updates. Most likely, they will not be the same. However, the committed IMS changes will be delivered to and committed in the DB2 side without any loss of IMS changed data.
- The points of consistency between IMS and DB2 are more difficult to achieve than if synchronous propagation is used. The CCU can not be used to verify consistency unless you make IMS database unavailable for updates during asynchronous updates. After the updates are applied to the target tables, you can run the CCU and then make the IMS databases available for update again.
- Error handling is more complex than with synchronous propagation.

Advantages and disadvantages of LOG-ASYNC propagation

LOG-ASYNC propagation supports only one-way IMS-to-DB2 propagation.

LOG-ASYNC propagation is propagation of changed data outside the commit scope, or UOW, of the IMS update. LOG-ASYNC propagation can occur in the same or a different MVS operating system, or in the same or different central processor complexes (CPCs), local or remote. If you have applications that cannot tolerate the performance impact of synchronous propagation, you might decide to use LOG-ASYNC propagation.

Advantages

The advantages of LOG-ASYNC propagation include:

- You can develop applications that use SQL to read the DB2 copy of the propagated IMS data.
- IMS and DB2 can reside on different MVS images. In this case, various data sets and libraries must be transported between the MVS images.
- You can operate your DB2 system independently of IMS. If problems develop on the DB2 side, you don't have to stop IMS; IMS is insulated from the DB2 problem.
- You can apply DB2 updates during periods of low machine activity.
- You can minimize performance impact on propagating applications.

Disadvantages

The disadvantages of LOG-ASYNC propagation include:

- The DB2 version of the propagated data might not reflect the latest version of the data.
- Neither one-way DB2-to-IMS nor two-way propagation are supported.
- There are no defined synchronization points or points of consistency between IMS and DB2. This can complicate database recovery efforts.

- Error handling is more complex than with synchronous propagation. When the propagation of a successful IMS update fails, there is no automatic backout of the IMS update. This makes it more difficult to keep the two data copies consistent.
- You might apply updates to the target tables in a sequence other than that in which they were made in IMS. Additional operational considerations and complexities in maintaining the proper sequence of updates can occur.
- Use of the CCU requires that IMS and DB2 have a common point of consistency. Therefore, the CCU cannot be used to verify consistency unless you make IMS databases unavailable for updates during asynchronous updates. After the updates are applied to the target tables, you can run the CCU and then make the IMS databases available for update.
- Combined use of DataRefresher and IMS DPROP, when DB2 and IMS are on different MVS images, is more difficult than when DB2 and IMS are on the same MVS image. For information on running LOG-ASync propagation on multiple MVS images see “When IMS and DB2 Reside on Different MVS Images” in *IMS DPROP Administrators Guide for Log Asynchronous Propagation*.
- If you implement user asynchronous propagation, you must write a receiver program to call RUP. Depending on how you implement user asynchronous propagation, you also need to write either a program to gather changed data from the log or a sender program.

Advantages and disadvantages of synchronous propagation

In order to compare synchronous to asynchronous propagation, this section focuses on the advantages and disadvantages of only one-way IMS-to-DB2 synchronous propagation.

During one-way IMS-to-DB2 synchronous propagation, when your applications update an IMS segment involved in propagation, the update is also applied to a corresponding DB2 table. This is done within the same UOW or commit scope as the IMS update. The synchronous propagation to DB2 appears to be part of the IMS update call. You can consider IMS DPROP an extension of the IMS application program. Both the application and IMS DPROP run in the same address space.

During the propagation phase, changes to IMS databases involved in synchronous propagation are collected by the IMS Data Capture function and passed to RUP, which runs as an exit to the Data Capture function. After examining the IMS DPROP directory, RUP performs mapping and data conversions. If you need to perform special mapping and conversions, RUP calls the IMS DPROP exit routines you have written. Then RUP calls the appropriate IMS DPROP-generated SQL update module to update the DB2 tables.

As with all IMS/DB2 mixed-mode applications, the IMS synchronization point manager handles the two-phase commit process. In this process, if a synchronous propagation failure occurs, the IMS synchronization point manager backs out the failing UOW. Consistency across IMS and DB2 components is maintained.

Synchronous propagation requires that the IMS and DB2 systems reside on the same MVS image. Synchronous propagation applies to block-level data sharing environments when the IMS subsystem updating the propagated data is on the same MVS image as the DB2 system.

Advantages

The advantages of synchronous propagation include:

- You can perform one-way IMS-to-DB2, one-way DB2-to-IMS, and two-way propagation.
With one-way IMS-to-DB2 synchronous propagation, you can develop applications that use SQL to read the DB2 copy of the propagated IMS data.
With DB2-to-IMS and two-way synchronous propagation, you can develop applications that use SQL to read *and update* the DB2 copy of the propagated IMS data.
- With IMS-to-DB2 synchronous propagation, propagated data is available in DB2 tables on a real-time basis; this is usually the best choice for operational applications.
- Synchronization problems between IMS and DB2 are less likely to happen. If an update fails on either side, backout is done on both IMS and DB2. The backout is handled by the IMS synchronization point manager. The two-phase commit protocol establishes points of consistency between IMS and DB2.
- Recovery and planning for recovery are easier with synchronous propagation than with asynchronous propagation, because there are fewer functions that can fail and that, therefore, need to be secured.
- Using the CCU to verify data consistency is simpler.
- The operational environment is simpler because you do not need data collection and transportation.

Disadvantages

The disadvantages of synchronous propagation include:

- IMS and DB2 must reside on the same MVS image.
- System problems can require that you stop propagation, transactions, or the systems. If you have stopped synchronous propagation and the propagated databases have been updated, you will need to resynchronize IMS and DB2 data before resuming.
- Your application performance might be affected.

Chapter 5. How IMS and DB2 relate

This chapter describes some basic principles of IMS and DB2 databases and shows how the two database structures compare to each other. You must understand the relationships between database structures before designing and implementing propagation.

Characteristics of IMS databases

This section reviews basic characteristics and features of IMS databases. For detailed information on this topic, see *IMS/ESA Administration Guide: Database Manager*.

IMS database structure

You can view an IMS database as a hierarchy, or inverted tree, of database segments. The top or first segment in the inverted tree is the root segment. Branching out from the root are dependent segments, called children. Children can also have branches. Each branch can have many children, and each child can have more than one occurrence of its type within a given database record. That is, there can be many single, specific child segments (occurrences).

The highest level segment at a branch is the parent of the child segments below it. The existence of child segments is dependent on the existence of a parent segment. No child can exist without a parent, and the root can have no parents, only children. A root segment can occur only once in a database record. A given database record can occur only once in a particular database.

The hierarchy is maintained by pointers that IMS places in the segments of the database record. These pointers chain the segments together and create the hierarchy of the database record. Application programming interface facilities in IMS present the database record to application programs in a hierarchical sequence.

A hierarchy is processed in the following sequence:

1. Top to bottom
2. Front to back
3. Left to right

These concepts and terminology are shown in Figure 27 on page 60.

Within a segment are fields that contain the data for that segment. For example, within a customer segment, there might be a field containing the customer account number, another field with the customer name, and another field with the account balance.

You can also use one of the fields in a segment as a *key field*. A key field stores the segments in a database record in a sequential order. For example, you could use a customer account number field as the key field. Then an application program could search or sort the data based on the content of the field.

IMS Database Hierarchy

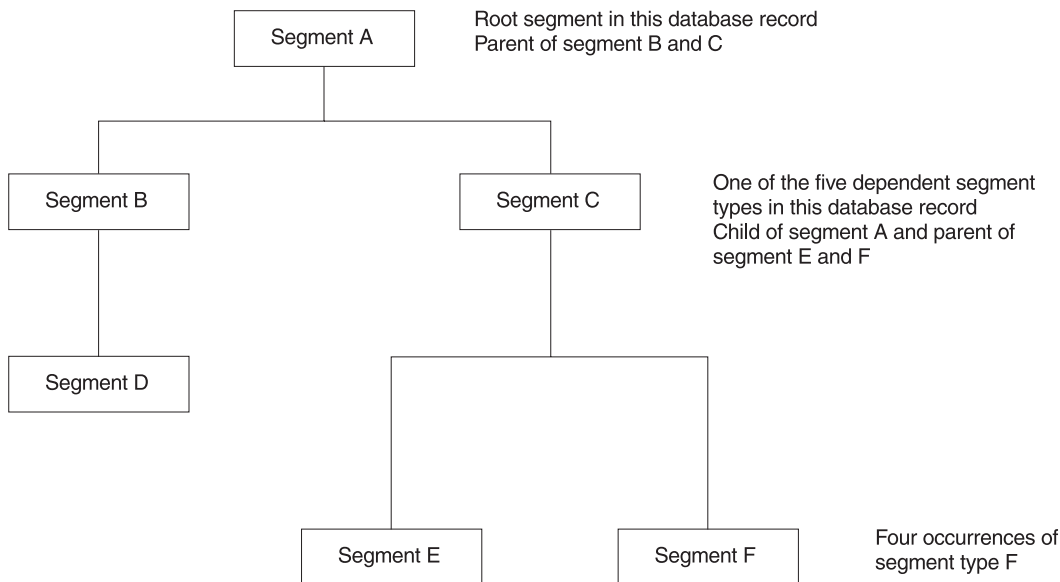


Figure 27. IMS Database Hierarchy

IMS logical databases

Defining logical databases enables you to combine physical databases with each other. When you define logical databases (relationships), your application program can:

- Access segment types in an order different from the one defined in the database hierarchy
- Access data structures in a single logical view that contains segments from multiple physical databases

If a segment participates in a logical relationship, the segment assumes the role of either a logical parent or logical child. With logical relationships, you can define sophisticated networks of segment types in addition to the original physical definitions of the databases. For more information on defining logical databases, refer to *IMS/ESA Application Programming: Design Guide* or *IMS/ESA Administration Guide: Database Manager*.

The concept of a logical database in IMS is shown in Figure 28 on page 61. The figure shows that segment A is a logical parent of segment N and a physical parent of segment B. Segment N is a logical child of segment A and a physical child of segment M.

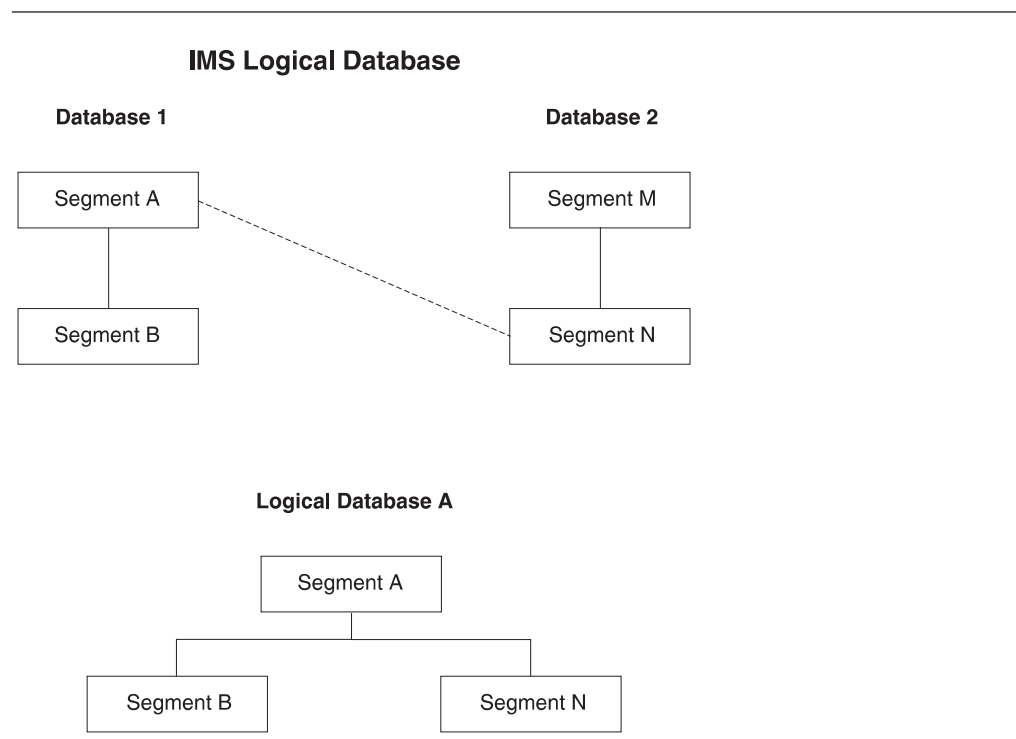


Figure 28. IMS Logical Relationships

IMS secondary indexes

Secondary indexes allow the segments of a database to be processed in a sequence other than the one defined by the segment's key. Secondary indexes can enforce the uniqueness of a field or of a combination of fields in the segment being indexed. Secondary indexes reside in a database separate from the one they are indexing. Because they are in a separate database, they can be processed as a database. For more information on secondary indexes, see *IMS/ESA Administration Guide: Database Manager*.

IMS database control blocks

A database description (DBD) is a control block that defines the physical implementation of IMS databases. The options that you specify in the DBD determine the collection of pointers that IMS constructs when the database record is built. A program specification block (PSB) is a collection of program control blocks (PCBs). PCBs define the view of the databases that an application program can access. An IMS database can reside on one or more database data sets (DBDSs).

IMS user interface

Data Language/One (DL/I) is the data manipulation language you use to interface to IMS databases. IMS DPROP does not restrict use of the various options provided by DL/I.

Characteristics of DB2 databases

This section reviews basic characteristics of DB2 databases. For detailed information on this topic, refer to the *DB2 Administration Guide*.

DB2 database structure

A DB2 database is a collection of DB2 tables, related indexes, and the table spaces. A table consists of a specific number of columns and any number of unordered rows. Within a table, all rows contain the same number of columns. The intersection points of these rows and columns can either have values or be null.

Note the similarities between the following DB2 and IMS concepts:

- DB2 tables and IMS segment types
- DB2 rows and IMS segment occurrences
- DB2 columns and IMS fields

You can implement parent/child relationships between DB2 tables that are similar to the parent/child relationships between IMS segment types. You do this by defining DB2 referential integrity relationships (RIRs) between tables.

When implementing an RIR between two tables, one table is considered the parent; the other table is considered the child.¹

- A primary key is defined in the parent table. The primary key is a column or group of columns that uniquely identifies each row in the table.
- A foreign key is defined in the child table. A foreign key is a column or group of columns in the child table.

The definition of the columns of the foreign key in the child table must match the definition of the columns of the primary key in the parent table. For example, row C in the child table is considered the child of row P in the parent table if the value of the foreign key of C matches the value of the primary key of P.

A record is how a row is represented in storage. In DB2, the application program does not need to know the actual structure of the record; DB2 insulates the application from this structure and presents to the application only the data defined in the application's view of the database.

DB2 indexes

Indexing in DB2 is similar to secondary indexing in IMS. However, unlike IMS indexes, DB2 indexes are transparent to the application program. A DB2 index is a set of pointers to data in a DB2 table, and is based on the value of data in one or more columns defined in a table. The index is stored separately from the rest of the table.

In DB2, indexes are used to improve performance and to ensure uniqueness. In some cases, you can access data more quickly with an index. In a table with a unique index, no two rows can have the same index. For more information on DB2 indexes, refer to the *DB2 Administration Guide*.

Relating indexes from IMS to DB2

If an index exists on a field in an IMS structure, you might not need to carry the index across to the propagated DB2 tables. You should analyze application access path requirements and then determine the index requirements in the target DB2 environment.

1. The DB2 terms are "dependent table" and "dependent rows", not "child table" and "child rows". However, IMS DPRDP books use the term "child" so that terms for IMS and DB2 are similar.

DB2 user interface

Structured Query Language (SQL) is the data manipulation and description language used in DB2 application program processing.

IMS and DB2 keys and their relationship

Database keys are the common links relating IMS segment occurrences and DB2 rows in a propagating environment.

Figure 29 on page 64 illustrates the IMS keys and the relationship to between databases.

Derivation of Concatenated Keys in IMS

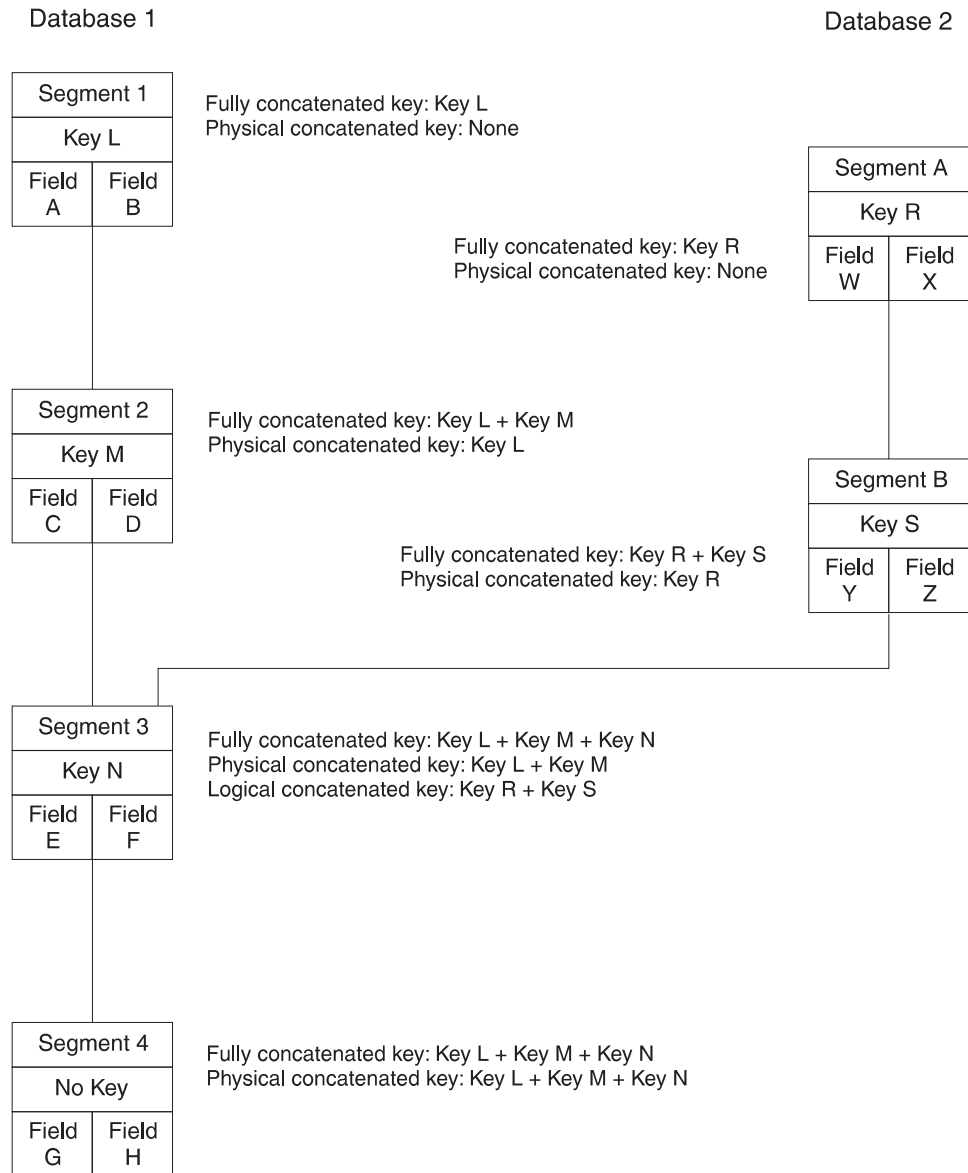


Figure 29. Derivation of Concatenated Keys in IMS. Database 2 is logically related to database 1; segment 3 is a logical child of segment B.

The three types of IMS keys shown in the Figure 29 are:

IMS key field

Usually IMS segments have a key field, although it is not required.

The IMS key field can be defined in the IMS DBD as unique or non-unique. If identified as unique, each occurrence of the segment under its physical parent has a different key field value.

The IMS key field is identified in the IMS DBD using the *SEQ* sub-parameter in the *FIELD* statement.

IMS fully concatenated key

For an IMS segment, the fully concatenated key consists of:

- The key field of the segment
- The key fields of the segment's physical parent and physical ancestors

IMS returns the fully concatenated key to applications in the key feedback area of the database's PCB when the segment is accessed through the physical path.

IMS concatenated key (physical and logical)

For an IMS segment, the concatenated key consists of the IMS key fields of the segment's immediate parent and ancestors.

Unlike the IMS *fully* concatenated key, the concatenated key does not include the IMS key of the segment itself.

DB2 referential integrity and propagation

DB2 lets you define referential integrity relationships (RIRs) between parent and child (dependent) tables.²

RIRs prevent update errors in relational databases. For example, DB2 RIRs prevent insertion of a row in a child table if there is no corresponding row in the parent table.

To implement a DB2 RIR, you define a primary key for the parent table and a foreign key for the child table. The value of the foreign key in a child row must match the value of the primary key in the parent row.

For IMS-to-DB2 propagation, use of referential integrity for propagated tables is optional, but supported by IMS DPROP.

For DB2-to-IMS and two-way propagation, use of referential integrity for propagated tables is strongly recommended.

If you implement RIRs, they should be compatible with the parent-child relationships of the propagated IMS segments. Propagation can fail if the referential integrity rules of the target DB2 tables do not match the parent/child relationship in IMS.

For details on implementing compatible DB2 RIRs, see the chapter titled "DB2 Referential Integrity Guidelines" in the appropriate *Administrators Guide* for the type of propagation that you are performing.

Representing IMS Structures in DB2

You can map IMS database structures and relationships to DB2 database structures and relationships. You can define an IMS segment type in DB2 as a table.

An IMS parent/child relationship (both logical and physical) can be expressed in DB2 using an RIR. Implement this in DB2 by:

- Designating the target table of the parent segment as the *parent table*.
- Designating the target table of the child segment as the *child table*.

2. To stress the similarities between IMS and DB2 relationships, this book uses the terms *child table* and *child row*, rather than the DB2 terms dependent table and dependent row.

- Defining a *primary DB2* key for each propagated table. Typically the DB2 primary key is defined as the propagated IMS fully concatenated key of the corresponding IMS segment.
- Defining a DB2 *foreign key* in the *child* table.

The foreign key must match the primary key of the *parent* table similar to the way the IMS physical or logical concatenated key of the child segment matches the fully concatenated key of its physical or logical parent segment.

Usually, the DB2 foreign key is defined as the propagated IMS physical/logical concatenated key of the IMS child segment. This is usually the same as defining the DB2 foreign key of a child table as the propagated IMS fully concatenated key of the physical/logical parent IMS segment.

Figure 30 shows the mapping of keys from IMS to DB2 and shows DB2 RIRS.

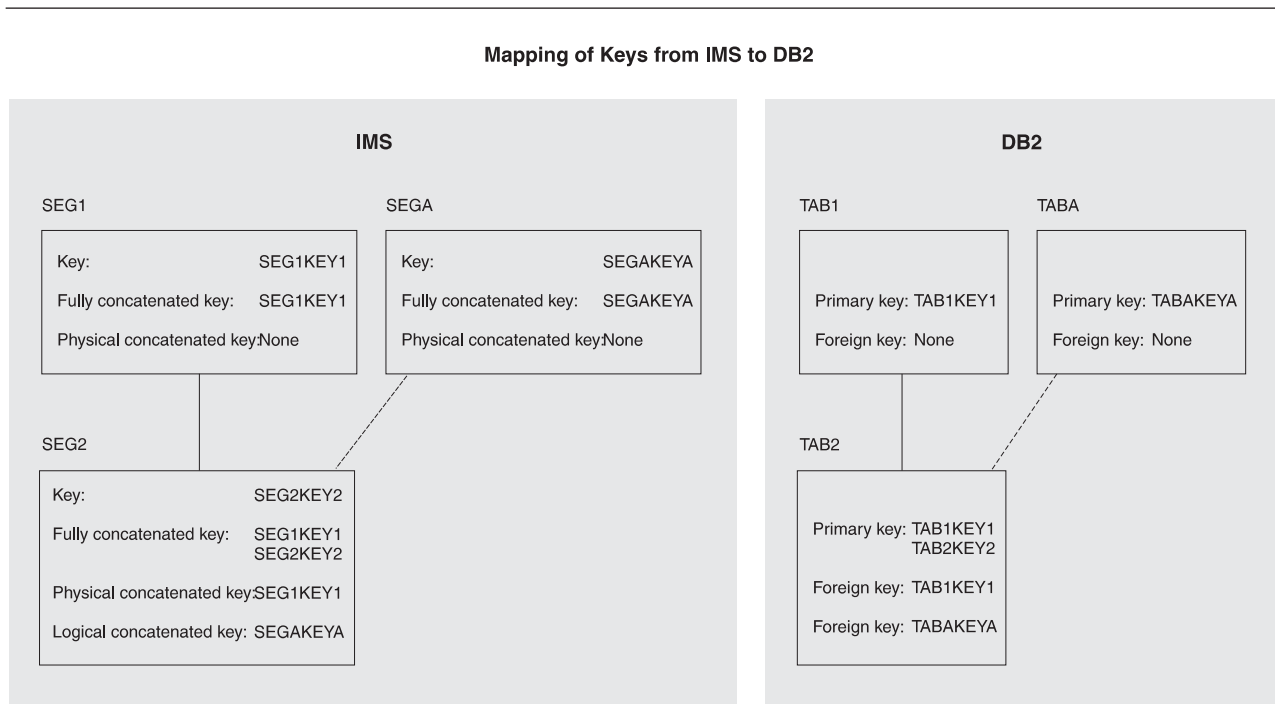


Figure 30. Mapping of Keys from IMS to DB2

In Figure 30, SEG1 is the physical parent of SEG2, and SEGA is the logical parent of SEG2. The DB2 primary keys are mapped from the IMS fully concatenated keys (SEG1KEY1 becomes, for example, TAB1KEY1). The DB2 foreign keys are mapped from the IMS physical concatenated key (SEG1KEY1 becomes TAB1KEY1) or from the IMS logical concatenated key (SEGAKEYA becomes TABAKEYA).

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, Program, or service may be used. Any functionally equivalent product, Program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described 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:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J74/G4
555 Bailey Avenue
P.O. Box 49023
San Jose, CA 95161-9023
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems.

Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

(c) (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. (c) Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information in softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This publication is intended to help you administer IMS DataPropagator, hereafter called IMS DPROP.

This publication also documents general-use programming interface and associated guidance information provided by IMS DPROP.

General-use programming interfaces allow the customer to write programs that obtain the services of IMS DPROP.

General-use 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:

Notice

This chapter documents general-use programming interface and associated guidance information.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AD/Cycle AT CICS	IMS Client Server/2
CICS/ESA CICS/MVS	IMS/ESA Information
COBOL/370 Database 2	Warehouse Language
DataPropagator	Environment MQSeries
DataRefresher DB2	MVS MVS/ESA OS/390
DFSMS DXT IBM IMS	QMF RACF SAA z/OS

Adobe, Acrobat, Portable Document Format (PDF), and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Glossary of Terms and Abbreviations

A

abort record. An IMS DataPropagator propagation log record (38nn or 5938), indicating that the associated unit of work will not be committed by IMS and should not be propagated to DB2. *Compare with commit record.*

ACB. Application control block. Located in IMS.

ACDC. Asynchronous changed data capture.

Apply Program. A component of IMS MQ-DPROP that reads the MQSeries messages containing the changed data and passes it to the RUP. RUP transforms the changed data into relational format and updates the DB2 target tables.

Archive utility. A utility that filters out propagation log records from the records written to the IMS logs and writes them to Changed Data Capture data sets (CDCDSs).

asynchronous changed data capture. An IMS function that captures the changes needed for IMS DPROP asynchronous propagation and saves them on the IMS logs. The function is mandatory for IMS DPROP asynchronous propagation and is either implemented by an SPE (IMS 3.1) or built into the program (subsequent releases of IMS).

asynchronous propagation. The propagation of data at a later time, not within the same unit of work as the update call.

Audit Extract utility. An IMS DPROP utility that inserts the IMS DPROP audit records written to SMF into the IMS DPROP audit table.

AUDU. Audit Extract utility.

B

Batch Log data set. A data set that an IMS batch job uses to store propagation log records needed for IMS DPROP asynchronous propagation.

C

CAF. Call attach facility.

CCU. Consistency Check utility.

CDCDS. Changed Data Capture data sets.

CDCDS Registration utility. An IMS DPROP asynchronous propagation utility that registers new CDCDS to DBRC.

CDCDS Unregistration utility. An IMS DPROP asynchronous propagation utility that deletes CDCDS entries from DBRC.

CDU. CDCDS Unregistration utility.

CEC. central electronics complex.

Changed Data Capture data set (CDCDS). The data sets that the archive utility uses to store the IMS DPROP asynchronous propagation log records filtered during the archive process. CDCDSs contain only the propagation log records. These log records are used by the Selector in place of the corresponding SLDSs, that contain all IMS changes.

Changed Data Capture exit routine. See DB2 Changed Data Capture exit routine

Changed Data Capture function. See DB2 Changed Data Capture function.

commit record. An IMS DPROP asynchronous propagation log record (9928, 37nn, 41nn, or 5937) indicating that the associated unit of work has been committed by IMS and should be propagated to DB2. *Compare with abort record.*

concatenated key. See “IMS concatenated key” and “conceptual concatenated key.”

conceptual concatenated key. The conceptual concatenated key of a segment consists of the concatenated keys of the segment's immediate physical parent and physical ancestors. Unlike the Conceptual *fully* Concatenated key, the conceptual concatenated key does not include the concatenated key of the segment itself.

conceptual fully concatenated key. The conceptual fully concatenated key is an IMS DPROP concept useful for the propagation of entity segments that do not have a unique IMS fully concatenated key; but that are nevertheless uniquely identifiable.

The conceptual fully concatenated key of a segment consists of these parts:

- the concatenated key of the segment
- the concatenated keys of the segment's physical parent and physical ancestors

The conceptual fully concatenated key is therefore the combination of these parts:

- the IMS fully concatenated key
- the ID fields (if any) of the segment that contribute to the concatenated key of the segment
- the ID fields (if any) of the physical parent or ancestors that contribute to the concatenated keys of the physical parent or ancestor

So, the conceptual fully concatenated key is equal to that hypothetical IMS fully concatenated key, that you would see if including the ID fields into the IMS key-field at each hierarchical level.

The concept of conceptual fully concatenated key allows the support of segments with a unique conceptual fully concatenated key, much in the same way as segments with a unique IMS fully concatenated key.

concatenated key. The concatenated key is an IMS DPROP concept useful for the propagation of entity segments that are neither unique under their parent nor have a unique IMS key, but that are nevertheless uniquely identifiable through ID fields.

The concatenated key is a combination of these fields that identify the segment uniquely under its parent:

- the non-unique IMS key field (if any)
- ID fields

For segments having a unique IMS key field, the conceptual key and the IMS key field are identical.

Consistency Check utility (CCU). An IMS DPROP utility that checks whether the data that has been propagated between IMS and DB2 databases is consistent. If not, it reports the inconsistencies and generates statements the DBA can use to fix the inconsistencies. The CCU is applicable when generalized mapping cases are being used.

containing IMS segment. An IMS segment that contains internal segments (embedded structures) propagated by mapping case 3 Propagation Requests. It is referred to interchangeably as a “containing IMS segment” or “containing segment.”

containing segment. See containing IMS segment.

CRU. CDCDS Registration utility.

D

Data Capture exit routine. See IMS data capture exit routine.

data capture function. An IMS function that captures the changes needed for data propagation.

DataRefresher. An IBM licensed program that lets you extract selected operational data on a periodic or one-time basis.

Data Extract Manager (DEM). A DataRefresher component that extracts the IMS data to which changes will subsequently be propagated. DEM also creates control statements for the DB2 Load utility to load the extracted IMS data into DB2 tables.

data propagation. The application of changes to one set of data to the copy of that data in another database

system. See also synchronous propagation and IMS DPROP asynchronous propagation.

DataRefresher DEM. DataRefresher data extract manager.

DataRefresher Map Capture exit routine (MCE). See Map Capture exit routine.

DataRefresher UIM. See User Input Manager.

DBRM. Database Request Module.

DB2 commit count. The number of IMS commit records that the IMS DPROP asynchronous propagation receiver is to apply to DB2 before it issues a DB2 commit.

DB2 Changed Data Capture exit routine. The routine to which the DB2 Changed Data Capture function passes the DB2 changes it has captured for propagation. This routine can be the IMS DPROP HUP routine, that propagates data, or your own exit routine.

DB2 Changed Data Capture function. A DB2 function that captures the DB2 changes needed for data propagation.

DB2 Changed Data Capture subexit routine. An optional IMS DPROP exit routine invoked whenever the HUP is called by DB2 changed data capture. The DB2 Changed Data Capture subexit routine can typically be used to perform generalized functions such as auditing all of the captured DB2 changes.

DB2-to-IMS propagation. Propagation of changed DB2 tables to IMS segments. It can be either:

- One-way DB2-to-IMS propagation
- DB2-to-IMS propagation, as part of two-way propagation

DBD. Database definition. The collection of macroparameter statements that describes an IMS database. These statements describe the hierarchical structure, IMS organization, device type, segment length, sequence fields, and alternate search fields. The statements are assembled to produce database description blocks.

DBDLIB. Database definition library.

DBPCB. Database program communication block.

DEDB. Data entry database.

DEM. Data Extract Manager.

directory. See IMS DPROP directory.

DLU. DL/1 Load Utilities. IMS DPROP utilities that are used to create (or re-create) the IMS databases from the content of the propagated DB2 tables. You can use DLU if you have implemented DB2 to IMS or two-way propagation.

DPROP-NR. The abbreviation for IBM IMS DataPropagator MVS/ESA through Version 2.2. At Version 3.1 the product name changed to IMS DataPropagator, abbreviated as IMS DPROP.

E

EKYMQCAP. The Capture component of MQ-DPROP. EKYMQCAP is an IMS data Capture exit routine. It runs as an extension to the updating IMS application programs, but it is transparent to them. EKYMQCAP obtains the changed data from the IMS Data Capture function and sends this data via MQSeries messages to the Apply Program.

EKYRESLB Dynamic Allocation exit routine. An IMS DPROP exit routine that can be used to allocate dynamically the IMS DPROP load module library to the EKYRESLB DD-name.

entity segment. The data being mapped from IMS to DB2 comes from one single hierarchic path down to a particular segment. This segment is called the entity segment. See also mapping case 1.

ER. Extract request.

Event Marker. A component of MQ-DPROP that runs on the same system as the IMS source databases. It is used to identify an event that occurs on the Source System. The customer must execute the Event Marker on the Source System at the time that the event occurs.

The Event Marker transmits an MQSeries message that identifies the event to the Apply Program. This MQSeries message is transmitted in FIFO sequence and in the same Propagation Data Streams as the changed IMS data.

When an occurrence of the Apply Program processes this message, the content of the target DB2 tables of this occurrence of the Apply Program reflect the content of the IMS source databases at the time that the Event Marker was executed on the Source System.

The Event Marker is used for an automated stop of the Apply Program when the content of the target DB2 tables reflects a particular Source System point in time.

exit routines. IMS DPROP contains seven exit routines. See the individual glossary entries for:

- DB2 Changed Data Capture exit routine
- DB2 Changed Data Capture subexit routine
- IMS Data Capture exit routine
- Field exit routine
- Map Capture exit routine
- Propagation exit routine
- Segment exit routine
- User exit routine

extension segment. The data being mapped from IMS to DB2 comes from a single hierarchic path down to an entity segment and from any segments immediately subordinate to the entity segment. The

segments subordinate to the entity segment can have zero or one occurrence beneath a single occurrence of the entity segment. This type of subordinate segment is called an extension segment (as it extends the data in the entity segment). See also mapping case 2.

extract request (ER). A DataRefresher request to extract IMS data. Extract requests become IMS DPROP propagation requests once they are validated by the IMS DPROP MCE.

F

Field exit routine. An IMS DPROP exit routine you can write to complement the logic of IMS DPROP's generalized mapping cases. Field exit routines are typically used to convert an individual IMS data field between a customer format IMS DPROP does not support and a format you have defined in your propagation request.

FIFO. First-In-First-Out

fully concatenated key. See IMS fully concatenated key and conceptual fully concatenated key.

G

generalized mapping cases. The mapping cases provided by IMS DPROP. See mapping case 1, mapping case 2 and mapping case 3.

group definition file. The file that the Group Unload utility (GUU) uses to store the IMS sources that it extracts from the IMS DPROP directory tables. *See also, SCF Compare job and SCF Apply job.*

Group Unload utility (GUU). The IMS DPROP asynchronous propagation utility that extracts details of all IMS sources for the specified propagation group from the IMS DPROP directory tables at the receiver site and writes them to the Group Definitions File. *See also, SCF Compare job and SCF Apply job.*

GUU. Group Unload utility.

H

hierarchical update program (HUP). The IMS DPROP component that does the actual DB2-to-IMS propagation. HUP is the IMS DPROP-provided DB2 Changed Data Capture exit routine. The DB2 Changed Data Capture function calls HUP and provides to HUP the changed IMS rows.

Hierarchical to Relational propagation. This is one-way hierarchical to relational propagation: the one-way propagation of changed IMS segments to DB2 tables. The terms *hierarchical to relational propagation* and *one-way IMS-to-DB2 propagation* are interchangeable.

HUP. Hierarchical Update program.

HSSR. High speed sequential retrieval.

I

ID fields. *Identification (ID) fields* are non-key fields that:

- uniquely identify a segment under its parent
- do not change their value

Typical examples of IMS segments with ID fields, are segments where the data base administrator has not defined the ID fields as part of the IMS Key field. For example because the IMS applications need to retrieve the segment in another sequence than the ascending sequence of the ID fields.

identification fields. See ID fields.

IMS concatenated key. For an IMS segment, the concatenated key consists of:

- The key of the segment's immediate parent, and
- The keys of the segment's ancestors

Unlike the IMS **fully** concatenated key of the segment, the concatenated key does not include the key of the segment itself.

A logical child segment has two concatenated keys: a physical concatenated key and a logical concatenated key. The physical concatenated key consists of the key of the segment's physical parent and the keys of the physical ancestors of the physical parent. The logical concatenated key consists of the key of the segment's logical parent and the keys of the physical ancestors of the logical parent.

IMS Data Capture exit routine. The routine to which the IMS Data Capture function passes the IMS changes it has captured for propagation. For synchronous propagation, this routine can be the IMS DPROP RUP routine, that propagates data, or your own exit routine. For IMS DPROP asynchronous propagation, the data capture exit routine is a program you write that gets the changed data from IMS. Other programs that you write will later invoke IMS DPROP with the changed IMS data.

IMS data capture function. An IMS function that captures the changes needed for data propagation.

IMS DPROP. The abbreviated name for the IBM IMS DataPropagator product. Previously, this product was called IMS DataPropagator, abbreviated as DPROP-NR.

IMS DPROP directory. A set of DB2 tables containing the mapping and control information necessary to perform propagation.

IMS fully concatenated key. For an IMS segment, the fully concatenated key consists of:

- The key of the segment,

- The key of the segment's immediate parent, and
- The keys of the segment's ancestors.

Unlike the IMS concatenated key of the segment, the fully concatenated key includes the key of the segment itself.

IMS INQY data. The first 9904 (update) record in each IMS unit of work (UOW) contains IMS INQY data (transaction name, PSB name, and user ID). This information is written to the PRDS for the propagation group as the first record of the UOW.

IMS log files. The files that IMS uses to store details of all changes to IMS data. See also, batch log data sets, online data sets (OLDSs), system log data sets (SLDSs), and Changed Data Capture data sets (CDCDSs).

IMS logical concatenated key. One of the two IMS concatenated keys of a logical child segment (the other is an IMS physical concatenated key). The logical concatenated key consists of:

- The key of the segment's logical parent, and
- The keys of the physical ancestors of the logical parent.

IMS physical concatenated key. One of the two IMS concatenated keys of a logical child segment (the other is an IMS logical concatenated key). The physical concatenated key consists of:

- The key of the segment's physical parent, and
- The keys of the physical ancestors of the physical parent.

IMS-to-DB2 propagation. This is the propagation of changed IMS segments to DB2 tables. Distinguish between:

- One-way IMS-to-DB2 propagation
- IMS-to-DB2 propagation, as part of two-way propagation

internal segments. Internal Segments is the IMS DPROP and DataRefresher term for structures embedded in IMS Segments, that are propagated through mapping case-3 propagation requests. Each embedded structure (i.e. each internal segment), is propagated to a different table; each occurrence of the embedded structure to one row of the table.

invalid unit of work. An IMS UOW that is missing a first record (containing the INQY data). If the IMS DPROP asynchronous propagation Selector detects an invalid unit, it responds according to what you specified on the INVUOW keyword of the SELECT control statements. If you specified:

IGNORE

The Selector continues processing

STOP

The Selector issues an error message and terminates

ISC. Inter-system communications.

ISPF. Interactive system production facility or Interactive structured programming facility.

IXF. Integrated exchange format.

L

LOG-ASYNC. The IMS log-based, asynchronous propagation functions of IMS DPROF.

Once the IMS log records are archived (IMS Online Logs) or de-allocated (IMS Batch Logs) by IMS and then stored in time-stamp sequence, LOG-DPROF reads the IMS logs to find the changed data and then stores the changed data in PRDS datasets. The Receiver component of IMS DPROF reads the PRDSs, transforms the data into the relational format, and applies the changes to the target DB2 tables.

See asynchronous propagation.

logical concatenated key. See IMS logical concatenated key

M

Map Capture exit (MCE) routine. The map capture exit routine provided by DPROF. MCE is used when you provide mapping information through DataRefresher. MCE is called by DataRefresher during mapping and data extract to perform various validation and checking operations. The IMS DPROF MCE should be distinguished from the DataRefresher Map Capture exit, the DataRefresher routine that calls MCE.

mapping case. A definition of how IMS segments are to be mapped to DB2 tables. IMS DPROF distinguishes between mapping case 1, mapping case 2, and user mapping cases.

mapping case 1. One of the generalized mapping cases provided by IMS DPROF. Mapping case 1 maps one single segment type, with the keys of all parents up to the root, to a row in a single DB2 table.

mapping case 2. One of the generalized mapping cases provided by IMS DPROF. Mapping case 2 maps one single segment type, with the keys of all parents up to the root, plus data from one or more immediately subordinate segment types (with a maximum of one occurrence of each segment type per parent), to a row in a single DB2 table.

mapping case 3. One of the generalized mapping cases provided by IMS DPROF. Mapping case 3 supports the propagation of segments containing embedded structures. A typical example of an embedded structure is a repeating group of fields.

- each embedded structure can be propagated to/from a different table. Mapping case 3 propagates each occurrence of an embedded structure, with the key of

the IMS segment, and the keys of the physical parent and ancestor, to/from a row of one DB2 table.

- the remaining data of the IMS segment (that is the fields that are not located in an embedded structure) can be propagated to/from another table.

Mapping Verification and Generation (MVG). An IMS DPROF component that validates the mapping information for each propagation request and stores it in the IMS DPROF directory. For a propagation request belonging to a generalized mapping case, MVG generates an SQL update module. MVG is invoked internally by MCE and MVGU.

Mapping Verification and Generation utility (MVGU). An IMS DPROF utility invoked by the DBA. MVGU creates propagation requests when DataRefresher is not used to provide mapping information (i.e., when you put the mapping information directly into the MVG input tables). MVGU also deletes or rebuilds propagation requests in the IMS DPROF directory.

master table. The IMS DPROF directory master table, that is created when IMS DPROF is initialized. It consists of one row, containing system and error information.

MCE. Map Capture exit routine.

MIT. Master Index Table.

MQ-ASYNC. The MQSeries-based, asynchronous propagation functions of IMS DPROF.

An IMS Data Capture Exit routine provided by IMS DPROF obtains the IMS Database changes in real time from IMS and sends the changes via MQSeries messages to an IMS DPROF Apply program. The Apply program reads the MQSeries messages, transforms the data into relational format, and then applies the new data to the target DB2 tables.

MQ-ASYNC supports both near-real time propagation and automated point-in-time propagation.

MQSeries. A family of IBM licensed programs that provide message queuing services.

MQSeries for OS/390. The members of the MQSeries that run on OS/390 systems.

MSDB. Main storage database.

MSC. Multisystem communication.

MVG. Mapping Verification and Generation.

MVG input tables. A group of DB2 tables into which the DBA stores propagation request definitions when DataRefresher is not used to provide mapping information. Once the propagation requests are stored, the DBA invokes MVGU. MVGU invokes MVG, that

validates the propagation request and copies the mapping definitions from the MVG input tables to the IMS DPROP directory.

MVGU. Mapping Verification and Generation utility.

N

Near RealTime. A delay of only a couple of seconds.

O

OLDS. Online Data Set.

One-way DB2-to-IMS propagation. This is the propagation of changed DB2 tables to IMS segments. Distinguish between:

- One-way DB2-to-IMS propagation
- DB2-to-IMS propagation, as part of two-way propagation

One-way IMS-to-DB2 propagation. This is the propagation of changed IMS segments to DB2 tables. Distinguish between:

- One-way IMS-to-DB2 propagation
- IMS-to-DB2 propagation, as part of two-way propagation

P

PCB. Program communication block.

persistent MQSeries message. An MQSeries message that survives a restart of the MQSeries Queue Manager.

physical concatenated key. See IMS physical concatenated key.

Point In Time Propagation. An Asynchronous propagation is said to operate in 'Point In Time' mode, when the data content of the target databases matches the content of the source databases at a previous, clearly identified Point In Time. For example, a Point In Time Propagation can be used to reflect in the content of the target databases the logical end of a business day, or the logical end of business month, or the end of specific Batch jobstream that updated the source databases.

PR. Propagation request.

PR ID. Propagation request identifier.

PRCT. Propagation Request Control Table

PRDS. Propagation Request Data Set

PRDS register file. A data set created by the IMS DPROP asynchronous propagation Selector that contains details of the associated PRDS.

PRDS register table. An IMS DPROP directory table that is created at the Receiver site when IMS DPROP is installed. The table is initially empty and you must populate it, using the PRU REGISTER control statements.

PRDS Registration utility (PRU). An IMS DPROP asynchronous propagation utility that registers PRDSs in the PRDS Register Table.

propagation. See data propagation.

Propagation Data Stream. A stream of changed IMS data that flows in MQSeries messages from the Capture Component of IMS DPROP to the Apply Component of IMS DPROP. Propagation data streams are defined with PRSTREAM control statements in the //EKYTRANS file of EKYMQCAP.

propagation delay. The time elapsed between the update of the IMS source database by the application programs and the update of the target DB2 table by IMS DPROP.

Propagation exit routine. An IMS DPROP exit routine you can write to propagate data when the generalized mapping cases don't meet your needs. A Propagation exit routine must provide all the logic for data mapping, field conversion, and propagation.

propagation group. A subset of the propagation requests in the IMS DPROP directory propagation request table (IMS DPROP asynchronous only).

You can define as many propagation groups as you like, but any propagation request can be associated with one and only one propagation group.

propagation log records. IMS log records that the IMS DPROP asynchronous propagation Selector writes to PRDSs:

- 9904 (update) records
- Commit or abort records
- SETS/ROLS records

propagation request control table (PRCT). An IMS DPROP directory table that is created at the Receiver site when IMS DPROP is installed. It contains details of all propagation requests defined to IMS DPROP and, in combination with the RCT, enables the Receiver to ascertain:

- Which propagation requests are assigned to which Receivers
- The activity status of all defined Receivers
- The activity status of all propagation requests that are assigned to defined Receivers

Propagation Request data set (PRDS). A sequential file into which the IMS DPROP asynchronous propagation Selector writes all propagation log records for a propagation group.

propagation request (PR). A request to propagate data between IMS and DB2. You define propagation requests for each segment type that is to be propagated.

PR set. A group of logically related propagation requests, identified by having the same PRSET ID. PR sets are typically used when you propagate the same IMS data to multiple sets of DB2 tables.

PRU. PRDS Registration utility.

PSB. Program specification block.

R

RCT. Receiver control table.

Receiver. An IMS DPROP asynchronous propagation component that retrieves the propagation log records from a PRDS and passes them to the RUP, that uses them to update the DB2 target tables.

Applies to LOG-DPROP.

RECEIVER control statement. A control statement that is input directly into the IMS DPROP asynchronous propagation Receiver JCL to specify:

- The name of the Receiver that is to process a PRDS
- The names of the DB2 subsystem to be accessed and the DB2 plan
- The number of committed UOWs to process before a DB2 commit is issued

Applies to LOG-DPROP.

Receiver control table (RCT). An IMS DPROP directory table, that is created at the Receiver site when IMS DPROP is installed. The table is initially empty and you must populate it, using the SCU CREATEREC control statement. It contains details of all Receivers and, in combination with the PRCT, enables the Receiver to ascertain:

- Which propagation requests are assigned to which Receivers
- The activity status of all defined Receivers
- The activity status of all propagation requests that are assigned to defined Receivers

Applies to LOG-DPROP.

Relational to Hierarchical propagation. This is one-way relational to hierarchical propagation: the one-way propagation of changed DB2 tables to IMS segments. The terms *relational to hierarchical propagation* and *one-way DB2-to-IMS propagation* are interchangeable.

relational update program (RUP). The IMS DPROP component that does the actual IMS to DB2 propagation. RUP is the IMS DPROP-provided IMS Data Capture exit routine.

For synchronous propagation, the IMS Data Capture function calls RUP with the changed IMS segments.

For user asynchronous propagation, your routine gets the changes from IMS and later calls RUP.

For IMS DPROP asynchronous propagation, the Receiver gets the changes from the Selector-Receiver Interface and later calls RUP. In either case, RUP propagates the changes to DB2.

RIR. RIR is an IMS DPROP abbreviation for DB2 Referential Integrity Relationship. Database administrators can define RIRs between tables in order to request that DB2 catches and prevents update anomalies in the relational databases.

Implementation of RIRs between propagated tables is:

- Optional for one-way IMS to DB2 propagation
- Strongly recommended for DB2 to IMS and two-way propagation

RTT. Resource translation table.

RUP. Relational Update program.

RUP control block table. A single IMS DPROP directory table that contains one RUP propagation control block (PRCB) for each propagated segment type. Each RUP PRCB contains details of the relevant database and segment.

S

SCF. Selector Control File.

SCF Apply job. Uses the SCF control statements to create new propagation groups and to list and modify existing propagation groups in the SCF.

SCF Compare job. Used to compare the contents of the Group Definitions File with the propagation groups in the SCF and to generate SCF control statements to bring the SCF into line with the Group Definitions File.

SCF control statements. Can be generated automatically by the IMS DPROP asynchronous propagation GUU or input directly into the IMS DPROP asynchronous propagation SCF Apply utility JCL. The control statements modify the contents of the SCF records.

SCU. Status Change utility.

segment exit routine. An IMS DPROP exit routine you can write to complement the logic of the generalized mapping cases. Segment exit routines are typically used to convert a changed data segment from the form it has in your IMS database to a form you have defined in your propagation request.

SELECT control statements. Control statements that are input directly into the IMS DPROP asynchronous propagation Selector JCL to define the execution options for the Selector.

Applies to LOG-DPROP.

Selector. An IMS DPROP asynchronous propagation component that collects propagation log records from the IMS log files and writes them to PRDSs for later processing by the IMS DPROP asynchronous propagation Receiver component.

Applies to LOG-DPROP.

Selector control file. Created at Selector installation or generation time and contains the following control information that is essential to the operation of the Selector:

- Database records and propagation group records
- DBRC information
- Timestamp information

Applies to LOG-DPROP.

SLDS. System Log Data Set.

SNAP. system network analysis program

Source System. An OS/390 system where IMS source databases of the IMS DPROP propagation reside.

SQL update module. A module generated by MVG for each propagation request belonging to a generalized mapping case. An SQL update module contains all the SQL statements required to propagate to DB2 the changed IMS data for that propagation request.

SSM. Subsystem member. An IMS JCL parameter that identifies the PDS member that describes connection between IMS and the DB2 subsystems.

Status Change utility (SCU). An IMS DPROP utility that:

1. Changes the status of propagation requests in the synchronous environment. Propagation requests can be active, inactive, or suspended. The SCU also performs a variety of other service functions.
2. Maintains the Timestamp Marker Facility and populates the RCT and the PRCT in IMS DPROP asynchronous propagation.

synchronous propagation. The propagation of data within the same unit-of-work as the update call.

T

Target System. An OS/390 system where DB2 target tables of the IMS DPROP propagation reside.

Timestamp Marker Facility. Supports the statements that create, assign, and delete timestamp markers in the SCF. It is run as part of the SCU.

TSMF. Timestamp Marker Facility.

TSMF Callable Interface. A facility that allows a user application to create a stop timestamp for one or more propagation groups.

Two-way propagation. The combination of IMS-to-DB2 propagation and DB2-to-IMS propagation for the same data.

TW propagation. See two-way propagation.

U

UIM. User Input Manager.

ULR. Uncommitted Log Record.

uncommitted log records (ULR). When the IMS DPROP asynchronous propagation Selector terminates, it writes all uncommitted log records (propagation log records that have not yet been either committed or aborted by IMS) to the uncommitted log record data set. On a subsequent Selector execution, these records will be either written to the appropriate PRDS (if they have been committed by IMS) or deleted from the uncommitted log record data set (if they have been aborted by IMS).

UOW. Unit of work.

USER-ASYNC. The User asynchronous propagation functions of IMS DPROP.

user exit. See exit routines.

User Input Manager (UIM). A DataRefresher component to which you describe your IMS databases and the mapping between IMS databases and DB2 tables. The mapping is defined by submitting extract requests. You can specify on an extract requests that the UIM is to invoke the DataRefresher Map Capture exit routine provided by IMS DPROP and pass it the DataRefresher mapping definitions of the extract request.

user mapping case. A mapping case you can develop if the generalized mapping cases don't meet your needs.

V

Virtual Lookaside Facility (VLF). An MVS/ESA component that is a specific implementation of data spaces. IMS DPROP exploits VLF for a high-performance retrieval of mapping information and other control information.

VLf. Virtual Lookaside Facility.

Bibliography

There are several other books in the IMS DPROP library. They are listed with their order numbers on the back cover of this book.

The following books are referred to in this book or might be helpful in understanding IMS DataPropagator for OS/390 and z/OS concepts:

SC26-3265	<i>DB2 Administration Guide</i> , Version 5	SC26-8013	<i>IMS/ESA Administration Guide: System</i> , Version 5
SC26-4891	<i>DB2 Command Reference</i> , Version 5	SC26-8034	<i>IMS/ESA Utilities Reference: Database Manager</i> , Version 5
SC26-3395	<i>DB2 Utility Guide and Reference</i> , Version 5	GN28-1257	<i>OS/390 MVS Application Development Guide</i>
SC26-3270	<i>DB2 SQL Reference</i> , Version 5	GC28-1473	<i>OS/390 MVS JCL User's Guide</i>
SC26-3269	<i>DB2 for OS/390 Data Sharing: Planning and Administration</i> , Version 4	GN28-1427	<i>OS/390 MVS Initialization and Tuning</i>
SC26-4248	<i>DXT Reference</i>	GC28-1449	<i>OS/390 MVS Setting up a Sysplex</i>
SC26-4636	<i>DXT Writing Exit Routines</i>	LY37-3710	<i>Remote Recovery Data Facility Program Description and Operation</i>
SH19-6999	<i>DataRefresher Command Reference</i>	GC28-1208	<i>OS/390 Parallel Sysplex Overview: An Introduction to Data Sharing and Parallelism</i>
SH19-6998	<i>DataRefresher Exit Routines</i>	GC26-3398	<i>An Introduction to DataPropagator Relational</i>
GC26-8744	<i>IMS Release Planning Guide</i> , Version 6		
SC26-8016	<i>IMS/ESA Application Programming: Design Guide</i> , Version 5		
SC26-8015	<i>IMS/ESA Application Programming: Database Manager</i> , Version 5		
SC26-8012	<i>IMS/ESA Administration Guide: Database Manager</i> , Version 5		
SC26-8020	<i>IMS/ESA Customization Guide</i> , Version 5		
SC26-8024	<i>IMS/ESA Installation Volume 2: System Definition and Tailoring</i> , Version 5		
SC26-8035	<i>IMS/ESA Utilities Reference: System</i> , Version 5		
SC26-8029	<i>IMS/ESA Operations Guide</i> , Version 5		
GC26-8031	<i>IMS Release Planning Guide</i> , Version 5		

Index

Special characters

&masy! propagation
when to use 5

A

accessibility x
active state 25
Apply component 30
Apply program
 IMS-to-DB2 propagation 33
 IMS-to-IMS propagation 33
Apply utility, SCF 34
asynchronous Data Capture function
 described 26
Asynchronous Data Capture function
 described 26
audit and trace information 37
Audit Extract utility 37
audit trail 37
AUDU (Audit Extract utility) 37

B

Base component 30
bibliography 81

C

CAF (call attachment facility)
 and DB2-to-IMS synchronous propagation 40
Call Attachment Facility (CAF)
 and DB2-to-IMS synchronous propagation 40
Capture component 30
Capture exit routine
 purpose 32
CCU (Consistency Check utility)
 checking for consistency 36
child row 65
child table 65
CICS (Customer Information Control System)
 restrictions 40
CICS attachment facility 40
coexistence of DL/I and SQL applications 51, 52
Compare utility, SCF 34
components of IMS DPROP 30
components, IMS DPROP 30
concatenated key, IMS 65
control blocks
 DBD 61
 IMS 61
 PCB 61
 PSB 61
COPY utility 18
Customer Information Control System (CICS) 40
 restrictions 40

D

Data Capture exit routine
 user-written 27
data sharing 37
database description
 IMS 61
database support, IMS 38
DataRefresher
 map capture exit 31
DB2
 attachment facility 40
DB2 Data Capture exit routine 35
DB2-to-IMS 8
DB2-to-IMS synchronous propagation 8, 18, 51, 53
 extract and load, description 18
 propagation phase 25
 RUP processing 35
DBD (database description)
 description 61
DBDLIB, IMS 41
deactive state 25
decision support applications 6, 7
DEDBs (data entry databases)
 sequential dependent segments 38
dependent row 65
dependent table 65
direction of propagation 1
disability x
DL/I (Data Language I) 61
DL/I Load utilities 35
DL/I repair statements 36
DLU (DL/I Load utilities)
 description 35
DXT vii
 terminology vii

E

environmental considerations
 IMS-to-IMS propagation 42
 synchronous propagation 40
 synchronous propagation with asynchronous
 propagation 41
 user asynchronous propagation 41
environments 39
 for MQ-ASYNCR propagation 39
 for receiver programs 41
 for the Receiver 39
 IMS DPROP 38
examples
 IMS-to-IMS propagation 43
exit routine
 Capture 32
exit routines
 Field exit routine 12
 general description 12
 Propagation exit routine 13

exit routines (*continued*)
 Segment exit routine 12
Extended Recovery Facility 40
extract and load 16, 17

F

fully concatenated key, IMS 65
functions of IMS DPROF 30

G

Group Unload utility 34
GUU (Group Unload utility) 34

H

hierarchical update program (HUP)
 description 35
hierarchy, IMS 59
HUP (hierarchical update program)
 description 35

I

IMS
 attachment facility 40
IMS Asynchronous Changed Data Capture function
 described 23, 24
IMS DPROF directory
 description 15
IMS DPROF phases
 description 30
IMS DPROF uses with IMS-to-DB2 propagation 51
IMS DPROF! phases 13
IMS DPROFcomponents 30
IMS-to-DB2 propagation 4, 17
 Apply program 33
 key mapping 66
 LOG-ASYNC propagation 23
 MQ-ASYNC propagation 21
 one-way, LOG-ASYNC propagation 56
 one-way, synchronous mode 57
 propagation phase 24
 RUP processing 34
 synchronous propagation 24
 user asynchronous propagation 26
IMS-to-IMS propagation 1, 16
 Apply program 33
 environmental considerations 42
 examples 43
 LOG-ASYNC with MQ-ASYNC propagation 20
 MQ-ASYNC propagation on different z/OS
 images 19
inactive state 25
indexes
 DB2 62
 IMS secondary 61
 relating from IMS to DB2 62
information warehouse 7

Information Warehouse 6
interfaces
 DB2 user 63
 IMS user 61

K

key field, IMS 59, 63
keyboard shortcuts x
keys
 IMS 63
 IMS concatenated 65
 IMS fully concatenated 65
 IMS logical concatenated 65
 IMS physical concatenated 65
 non-unique key field 64
 relationship of IMS and DB2 63
 unique key field 64

L

loading data
 into IMS databases 35
log-async propagation
 when to use 7
Log-ASYNC propagation
 using the IMS Asynchronous Changed Data Capture
 function 23
LOG-ASYNC propagation 51
LOG-ASYNC propagation
 advantages 56
 description 56
 disadvantages 56
 environmental considerations 39
logical concatenated key 65
logical databases 60
LOGICAL delete rule 38
logical relationships
 IMS DPROF support of 38
LookAt message retrieval tool ix

M

maintenance and control phase
 description 30
map capture exit 31
mapping
 definitions with utilities 31
 user 11
mapping and design phase 13, 15
mapping cases 10
mapping options 11
mapping verification and generation 31
mapping verification and generation (MVG) input
 tables 14
mapping verification and generation utility 31
MCE (map capture exit)
 description 31
message retrieval tool, LookAt ix
migration 52

- modes of IMS-to-DB2 propagation
 - asynchronous versus synchronous 55
- modes of propagation 1
- modified IMS archive job
 - filtering from propagation log records 23
 - writing to CDCDSs 23
- MQ-ASYNC propagation
 - advantages 55
 - disadvantages 56
 - environmental considerations 39
- MQ-ASYNC! propagation
 - description 55
- MVG (mapping verification and generation)
 - description 31
- MVG input tables
 - description 14
- MVGU (Mapping Verification and Generation utility) 31

N

- non-unique key field 64

O

- one-way
 - IMS-to-IMS propagation 1
- one-way IMS-to-DB2 propagation
 - LOG-ASYNC propagation 56
 - MQ-ASYNC propagation 55
- one-way IMS-to-DB2 synchronous propagation 51, 53
- operational and decision support applications 52
- operational applications 7, 51, 53

P

- parent table 65
- PATH data 11
- PCB (program control block) 61
- phases of propagation 13
 - description 30
- physical concatenated key 65
- PHYSICAL delete rule 38
- PR (propagation request)
 - changing the status of 35
- program control block 61
- program specification block 61
- programs
 - HUP 35
 - receiver 26
 - RUP 34
 - selector 26
 - sender 27
- propagation
 - direction 1, 10
 - modes 1
 - MQ-ASYNC advantages 55
 - MQ-ASYNC disadvantages 56
- propagation mode
 - asynchronous versus synchronous 58
- propagation phase
 - description 18

- propagation phase (*continued*)
 - IMS-to-DB2 LOG-ASYNC propagation 23
 - IMS-to-DB2 MQ-ASYNC propagation on different z/OS images 21
 - IMS-to-DB2 user asynchronous propagation 26
 - IMS-to-IMS LOG-ASYNC with MQ-ASYNC propagation on the same z/OS images 20
 - IMS-to-IMS MQ-ASYNC propagation on different z/OS images 19
 - synchronous propagation 24
- propagation request (PR)
 - description 15
 - status 35
- propagation states
 - during synchronous data propagation 25
- PSB (program specification block) 61

R

- Receiver component 30
- receiver program 26
- referential integrity relationship 65
- Remote Recovery Data Facility
 - description 29
- remote tables 40
- repair statements 36
- restrictions
 - environmental 38
- RIRs (referential integrity relationships)
 - description 65, 66
- RRDF (Remote Recovery Data Facility) 29
- RUNSTATS utility 18
- RUP (relational update program)
 - description 34

S

- SCF
 - Apply utility 34
 - Compare utility 34
 - Group Unload utility 34
- screen readers and magnifiers x
- SCU (Status Change utility)
 - description 35
 - performing tracing 36
- SDEP (sequential dependent) segment 38
- secondary indexes
 - description 61
 - IMS DPROP support of 38
- Selector component 30
- selector program 26
- setup phase
 - description 16
- SQL (Structured Query Language) 63
 - update module 15
- SQL repair statements 36
- structured query language 63
- structures
 - of DB2 databases 62
 - of IMS database 59
 - representing IMS structures in DB2 65

- suspended state 25
- Synchronous component 30
- synchronous propagation 8
 - advantages 58
 - description 57
 - disadvantages 58
 - environmental considerations 40
 - propagation states during 25
 - when to use 7
- synchronous propagation with asynchronous propagation
 - environmental considerations 41
- Sysplex
 - and data sharing 37
 - documentation references for set up 37
 - how it works in IMS DPROP 37
 - shared control objects 37

T

- tables
 - IMS DPROP directory 15
 - MVG input
 - mapping and design phase 14
 - parent 65
 - remote 40
- trace and audit information 37
- tracing
 - description 36
- trademarks 68
- TSO attachment facility 40
- turned off state 25
- two-way synchronous propagation 9, 51, 53

U

- unique key field 64
- user asynchronous propagation 51, 53
 - alternative implementations 29
 - environmental considerations 41
 - IMS-to-DB2 8
 - using the IMS Asynchronous Data Capture function 26
 - using your Data Capture exit routine 27
- user interfaces
 - DB2 63
 - IMS 61
- user mapping 11
- uses of IMS DPROP with IMS-to-DB2 propagation 51
- utilities
 - AUDU (Audit Extract utility) 37
 - CCU 36
 - checking for consistency 36
 - COPY 18
 - DLU 35
 - MVGU 31
 - RUNSTATS 18
 - SCU (Status Change utility) 35
- Utilities component 30

V

- Virtual Lookaside Facility 24, 26
- VLF (Virtual Lookaside Facility)
 - description 24, 26

W

- WHERE clause 11

X

- XRF (Extended Recovery facility) 40



File Number: 5655-E52

Printed in USA

SC27-1544-01

