



**Mainstar Solutions for
Cloning IMS Systems and Databases**

**Ron Hauptert
Kelly Smith**

**Rocket Software, Inc.
© 2008**

Table of Contents

Table of Contents.....	2
List of Figures.....	2
Cloning IMS Systems	4
Refreshing IMS Databases	6
Database Refresh to a Cloned IMS System Skeleton.....	8
IMS Cloning and Refresh Use Case Examples.....	10
Creating IMS Application Test and Quality Assurance Environments.....	10
Offloading IMS Pointer Checker Utility.....	11
Creating Read-only IMS Environments.....	12
Creating IMS Reporting Databases.....	12
IMS Staging for Data Warehouse Loading.....	13
IMS Application and System Maintenance.....	13
Testing IMS Application Maintenance.....	13
Testing IMS System Maintenance.....	13
Testing New IMS Release Migration Procedures.....	14
Copying HALDB Databases.....	14

List of Figures

Figure 1 Clone and Rename for IMS Processing Steps Used to Clone an IMS System.....	5
Figure 2 Rapid Database Refresh for IMS Processing Steps to Refresh an IMS Database.....	7
Figure 3 Refreshing IMS Databases in a Cloned IMS System Skeleton.....	9

IMS is the preferred database management system for many enterprise applications that require high performance and high availability. These critical applications depend on IMS databases that respond quickly and are always available. Cloning IMS systems provides opportunities to offload some business and data management processes from production IMS systems to cloned copies. The offloaded processes can be run in parallel without affecting production IMS systems and applications.

Cloning IMS systems allows database administrators, application developers, and quality assurance testers to use production IMS data without adversely affecting the performance and availability of sensitive IMS business applications. IMS application development and test environments can be created easily and quickly without affecting production IMS application performance or availability.

IMS maintenance utilities are typically run on production data to ensure the ongoing health of the database system. Running routine IMS database maintenance utilities on production IMS data can impact IMS performance and reduce availability of critical IMS applications. Offloading routine database maintenance utilities such as IMS pointer checker to a cloned copy of an IMS system preserves production performance characteristics and allows full IMS pointer checker utility validation to be run in parallel with production applications.

Some specific uses for cloned IMS data include:

- Creating and refreshing IMS application test and quality assurance environments.
- Using cloned IMS systems to offload IMS pointer checker utility.
- Reduce production performance impact of read-only users by directing them to an IMS clone copy. This is particularly useful for creating reporting databases or staging databases to offload data warehouse loading.
- IMS application and system maintenance.
- Copying HALDB databases quickly and easily.

Mainstar's **Clone and Rename for IMS** (ICR) provides automation facilities to clone an entire IMS system and its data in minutes versus the hours it would take using traditional IMS cloning processes. **Mainstar's Rapid Database Refresh** (RDR) is an add-on option to ICR that provides automation to refresh select databases to create offload processing environments from production data. ICR and RDR can use storage-based fast-replication to copy data quickly and effectively while preserving host CPU and I/O resources. ICR and RDR support IBM FlashCopy, EMC TimeFinder, and Hitachi ShadowImage fast-replication services.

Cloning IMS Systems

Clone and Rename for IMS is the foundation for cloning production IMS systems. It provides automation to clone IMS data and to perform meta-data management functions to allow the data to be used by a cloned IMS system on the same or another shared disk System z logical partition (LPAR). It supports both data sharing and non-data sharing IMS environments. Cloned IMS systems can be used to offload application development, quality assurance testing, or database health check utilities such as the IMS pointer checker utility from a production IMS environment to a cloned IMS environment.

Clone and Rename for IMS leverages storage system capabilities and storage-based fast-replication to copy the IMS data quickly and effectively without using host CPU or I/O resources. Modern storage processing systems provide fast data replication and copying services that allow database data to be replicated instantaneously. Using storage processor functions to replicate the data on behalf of the database system reduces copy time and administration costs, and allows for an entire IMS system to be cloned or specific IMS databases to be refreshed in minutes while preserving valuable host processing resources.

ICR provides the automation required to use the replicated data in a target IMS system. It coordinates the replicated volume identifiers and replicated dataset names with their associated target IMS system and its meta-data to automate and simplify the IMS system cloning process. ICR performs fast and effective volume reconditioning and data set renaming operations to allow the replicated data to be used by a cloned IMS system on the same or another shared disk LPAR.

Cloning IMS data using storage-based fast-replication takes only seconds to replicate the data. Generally, complete IMS systems can be cloned in less than 15 minutes and databases can be refreshed in less than a minute. Figure 1 depicts the processing steps ICR uses to clone an IMS system. The steps shown in blue are performed on the production IMS system; steps shown in green are performed on the cloned data.

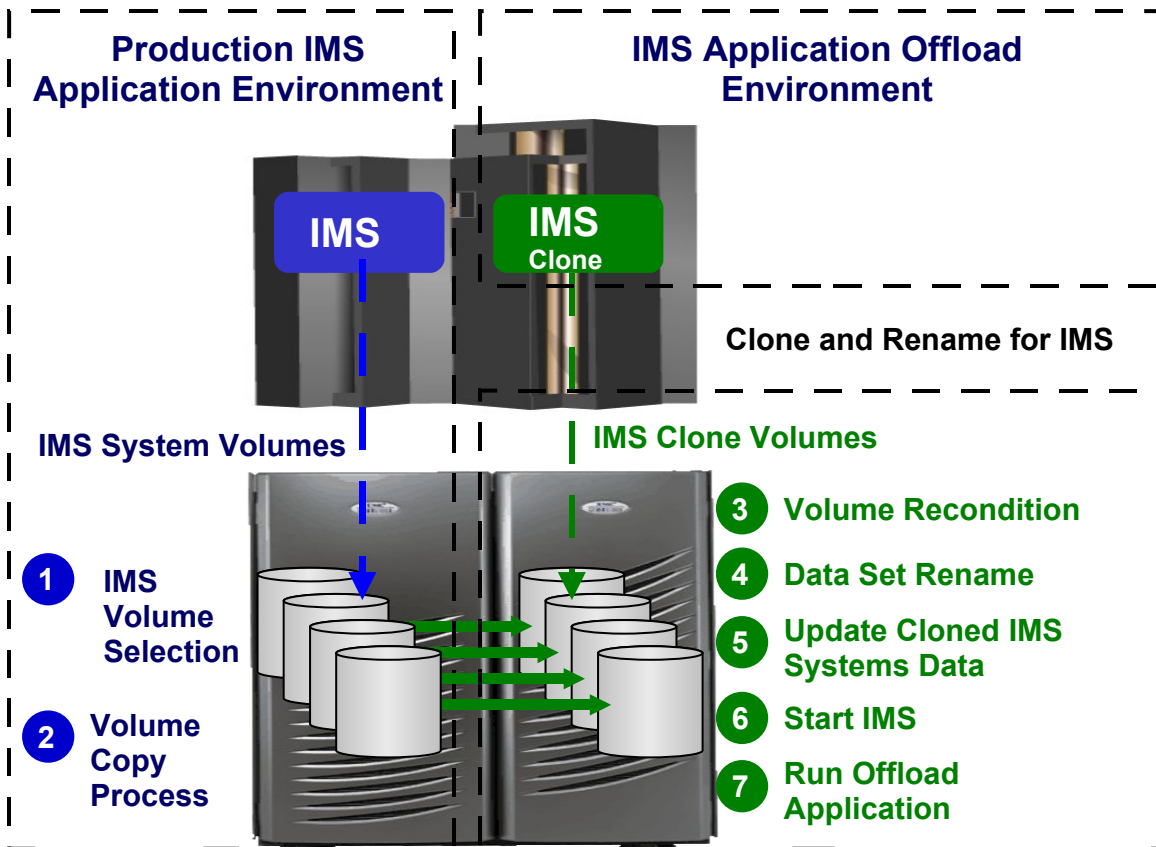


Figure 1 Clone and Rename for IMS Processing Steps Used to Clone an IMS System

1. The DASD volumes that make up the production IMS system and associated databases are identified by using specific VOLSER IDs, VOLSER masks, or SMS storage groups.
2. ICR invokes a DFSMSdss Copy to issue IBM FlashCopy or SnapShot commands to copy the data and backup up the ICF catalogs that point to the data sets on the volumes being copied. When cloning IMS systems that reside on EMC or Hitachi Storage Systems, an appropriate storage-based fast-replication process is performed before the ICR cloning automation is invoked and a list of copied storage volumes is passed to ICR for use in later processing steps.
3. If the target volumes are not already online, the volume internal identifier of each target volume is re-labeled so they can be brought online to the same z/OS system, or a different z/OS system, without volume label conflicts.
4. The data sets on the cloned volumes are renamed and re-cataloged to a new high level qualifier. Data sets are renamed to eliminate duplicate data set names from a z/OS catalog perspective and to allow the new data set names to be integrated into the cloned IMS system. The cloned and renamed IMS data sets can then be used on the same LPAR but integrated into a different IMS system using the new data set names.
5. The cloned IMS system is updated to reflect the new data set names, VOLSERS, IMSID, etc. The cloned IMS system updates include:

- a. RECON Data Sets - Data set names, IMS subsystem IDs, and VOLSERS are updated in the following RECON records: header record, database data set records, online log records, and back-out records.
 - b. Optionally, the following RECON records are updated: image copy records, change accumulation records, System Log Data Set (SLDS) records, and Recovery Log Data Set (RLDS) records if they were on volumes that were cloned.
 - c. IMS.PROCLIB, IMS.JOBS, and user JCL library members are updated with the new values for IMSID, VOLSERS, and data set names.
 - d. MDA (MVS Dynamic Allocation) members for databases or system data sets, such as the RECON, Online Log Data Sets (OLDS), or Write-Ahead Data Sets (WADS) are updated to reflect the new data set names.
 - e. If IMS data sharing is involved, each additional IMS data sharing member is updated.
6. The cloned IMS system is started and made available to users.
 7. Perform offload processing activities on the cloned IMS system. See “IMS Cloning and Refresh Use Case Examples” section.

Refreshing IMS Databases

Sometimes application environments require the ability to refresh specific IMS databases from a production IMS system. These databases can be used to offload application testing, quality assurance, create staging for data warehouse loading, etc. Having the ability to refresh specific databases from a production IMS system quickly can save time and resources over having to clone a complete IMS system.

Mainstar’s Rapid Database Refresh for IMS (RDR) is an ICR option that provides facilities to refresh databases within or across IMS systems in seconds. The target IMS database definition must exist on the target IMS system before databases can be refreshed. RDR assures the compatibility of the database definitions on the source and target IMS systems prior to initiating the data replication process. RDR can refresh IMS databases while they are stopped or while they are running to create fuzzy copies.

RDR can use storage-based data set level fast-replication to replicate production IMS data to a target IMS system. Production IMS data set names are changed on the refreshed copy during the fast-replication process to associate them to a target IMS system. Target IMS meta-data is updated to reflect the refreshed databases. Refreshing IMS databases using storage-based fast-replication takes only seconds to replicate the data and less than a minute to complete the database refresh process.

Figure 2 depicts the steps RDR uses to refresh databases from a production IMS system to an IMS application offload environment. Steps shown in green are performed by ICR’s Rapid Database Refresh for IMS option.

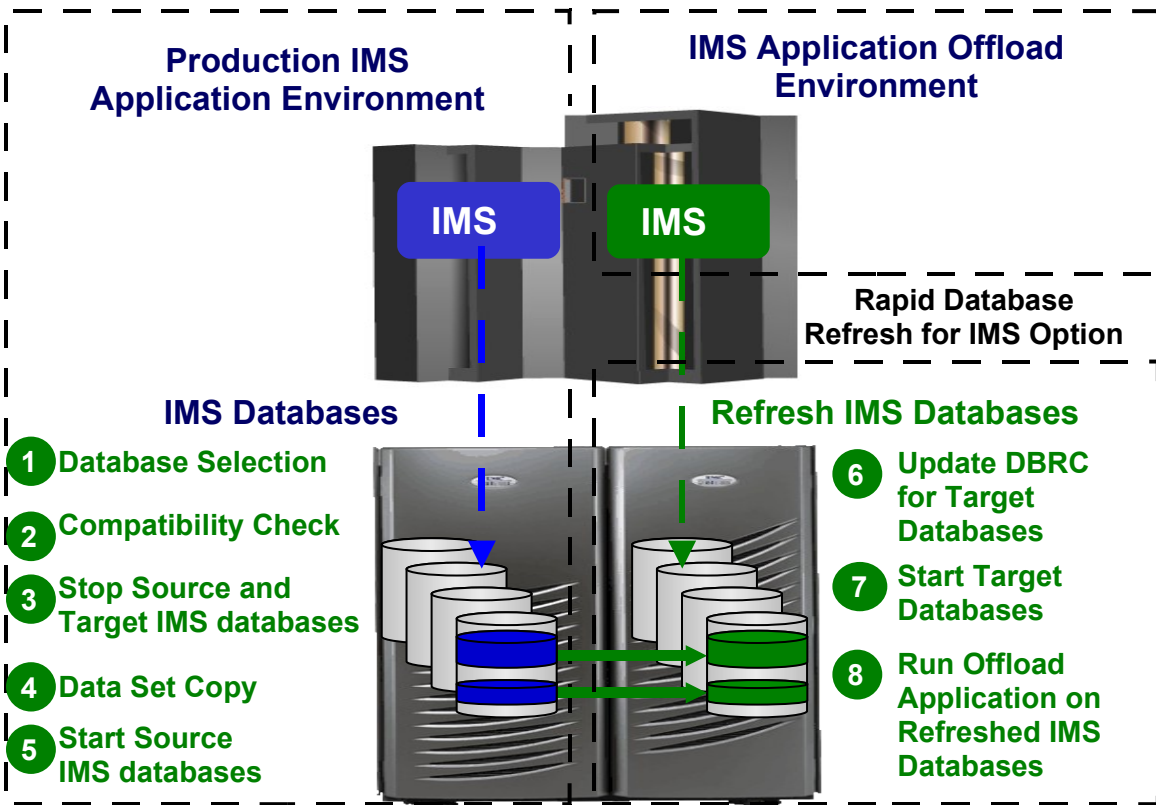


Figure 2 Rapid Database Refresh for IMS Processing Steps to Refresh an IMS Database.

1. The databases to be refreshed are selected on the source IMS system by database name. RDR determines all indexes associated with the databases to be refreshed and it determines all associated data set names for the specified databases.
2. Checks are performed to ensure the characteristics of the source and target IMS databases are compatible. Some characteristics that are checked include: database type, access method, number of segments, data set groups, blocksize, randomizer parameters, number of partitions, partition selection routine, KEYSTRNG, etc.
3. An IMS DBR command is issued for the selected target databases and indexes to stop them, and then a DBR command is issued for the source databases to create a consistency point. Optionally, the source objects can be copied while they are running to create a fuzzy copy. Care must be taken when using the fuzzy copy option as transactional integrity is not guaranteed on the cloned copy.
4. Data sets representing the database to be refreshed are copied. For IMS systems using FlashCopy or SnapShot, RDR will invoke these data set copy facilities from the RDR process. For IMS systems using EMC TimeFinder or Hitachi ShadowImage, or any other copy mechanism, RDR optionally produces files describing source and target data set information along with necessary control parameters so users can create their own data set fast-replication job streams.
5. The source IMS databases are started unless a fuzzy copy was specified in step three.
6. If the databases are registered to Database Recovery Control (DBRC), RDR updates the target IMS RECONS to prevent invalid recovery processes and sets

the image copy recommended flag on for the refreshed databases and indexes. RDR performs a LIST.DB to extract the RECON information and executes a DELETE.DB to remove the recovery history. RDR then redefines the refreshed databases and data sets by performing INIT.DB, INIT.DBDS, INIT.PART, and INIT.AREA operations.

7. Target IMS databases are started and made available to the IMS application offload environment.
8. Perform offload application processing activities on the refreshed IMS database(s). See “IMS Cloning and Refresh Use Case Examples” section

Database Refresh to a Cloned IMS System Skeleton

Cloning an entire IMS system and all associated database data can require significant storage resources when only some of the database data is of interest. Some IMS cloning applications may require only select database data to be cloned with the IMS system. A cloned IMS system skeleton can be created such that the cloned IMS system contains all database definitions but the database data is not initially copied. The database data is copied when a database refresh operation is performed on particular databases after the cloned IMS skeleton is created. Refreshing database data into a cloned IMS system skeleton can save storage resources and allow the database data to be refreshed in the cloned IMS system at different times and intervals.

A cloned IMS system skeleton is a cloned IMS system that contains all necessary data sets required to start and run an IMS system. It does not include the database data in the source IMS system. IMS system data sets required for the skeleton IMS clone include:

- IMS RESLIB and other data sets needed to get IMS up and running.
- IMS.PROCLIB, IMS.JOBS and user JCL libraries.
- ACB libraries (Application Control Blocks).
- OLDS (Online log data sets) and WADS (Write ahead data sets).
- MDA libraries.
- RECON data sets.

Figure 3 shows the creation of a cloned IMS system skeleton. Steps shown in blue are performed by ICR using production IMS system volumes. Steps shown in green are performed by RDR to populate the cloned IMS system by refreshing databases.

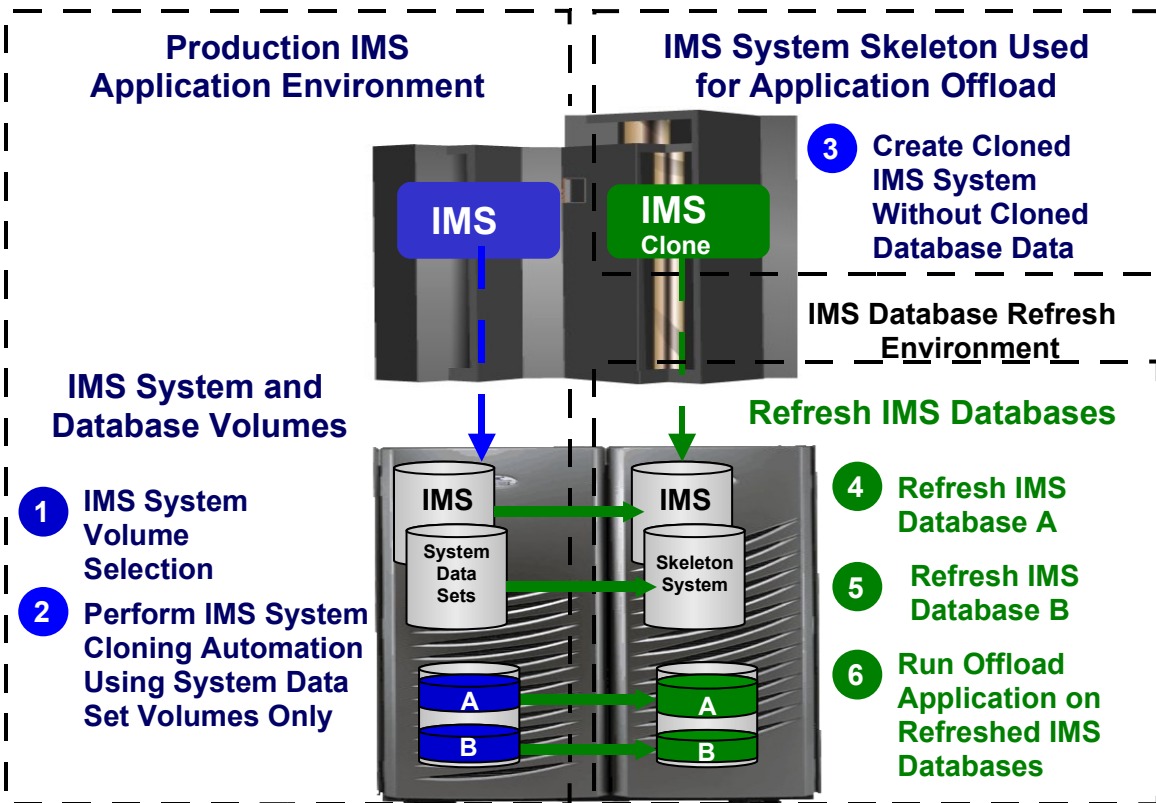


Figure 3 Refreshing IMS Databases in a Cloned IMS System Skeleton

1. Identify the volumes that contain IMS system data sets. This is done by identifying the volumes that contain the IMS system dataset referenced above.
2. Perform ICR IMS system cloning using the volumes identified in step 1. IMS system cloning steps are shown in figure 1.
3. The system cloning operation creates a usable IMS system with all databases defined but without database data.
4. Perform RDR database refresh operations for database A. This process can be run any time after the cloned IMS skeleton is created. IMS database refresh steps are shown in figure 2.
5. Perform RDR database refresh operations for database B. This process can be run any time after the cloned IMS skeleton is created and databases can be refreshed as frequently as desired.
6. Perform offload processing activities on the cloned IMS system and refreshed databases. See “IMS Cloning and Refresh Use Case Examples” section.

Once a cloned IMS skeleton is created, database data can be cloned from a production IMS system using the procedures described in the Refreshing IMS Databases section.

Having all IMS system datasets isolated on a dedicated set of volumes simplifies creating a cloned IMS skeleton and it alleviates issues associated with updating a cloned IMS system without affecting the cloned IMS database data. When IMS system data sets are collocated with IMS database data, the IMS system datasets and the collocated IMS database data are cloned together during the volume copy process, figure 1, step 2. The

volume copy process will overlay the cloned database data which may or may not be the intent of the operation.

IMS Cloning and Refresh Use Case Examples

IMS systems can be cloned to create IMS application offload environments. These offload application environments can be used to create and refresh test databases, run data quality assurance processes, allow read-only user access IMS data, to test new IMS maintenance, releases and migrations, and so forth. Creating IMS application offload environments by cloning IMS systems and refreshing IMS databases reduces the performance impact these applications have on production IMS systems. Some specific IMS system cloning and database refresh use case examples include:

- Creating IMS application test and quality assurance environments.
- Using cloned IMS systems to offload IMS pointer checker utility.
- Reducing production performance impacts of read-only users by directing them to an IMS clone copy. This is particularly useful for creating reporting databases or staging databases to offload data warehouse loading.
- Testing IMS application and system maintenance.
- Copying HALDB databases quickly and easily.

Creating IMS Application Test and Quality Assurance Environments

IMS applications evolve as new functions and features are added to support new business processes and procedures. Test IMS database environments are needed to develop and test new application functionality. Cloned IMS systems and refreshed IMS databases allow application developers to code and test using real IMS business data easily. All application development and testing processes can be performed using a cloned IMS system and refreshed database data to eliminate any adverse affect on production systems.

Quality assurance testing is critical to maintain high quality business operations. Quality assurance testing should be performed on production data to ensure accurate test results. Performing quality assurance testing on cloned IMS systems allows thorough application and data testing to be performed at will using production data without adversely effecting production systems.

IMS application development and quality assurance test environments can be offloaded to a cloned IMS system using the steps described in figure 1. Application development and testing processes are run in step number 7. IMS databases can be refreshed in the cloned IMS system using the steps described in figure 2.

Alternatively, a cloned IMS system skeleton can be created using the steps described in figure 3 and application development and quality assurance test environments can be offloaded to a refreshed IMS database using the steps described in figure 2. Application development and testing processes are run in step number 8.

Offloading IMS Pointer Checker Utility

The health of IMS databases is critical to applications and business processes that depend on them. These critical databases require a highly available support infrastructure and management framework to maintain high performance, high availability, and to minimize application outages. The database management framework and support infrastructure must include facilities to monitor backup and recovery capabilities, identify pointer errors that may impact database access, determine when to reorganize databases, perform analysis of corrupt databases, and monitor database usage and growth, and more. IMS database administrators depend on these support infrastructure facilities to maintain the health of IMS databases.

Most companies use the IMS pointer checker utility in their support infrastructure. The IMS pointer checker utility is run on a regular basis to monitor the database, identify potential problems, and minimize application outages. Using the IMS pointer checker utility can impact production database availability caused by running it when the database is stopped. The IMS pointer checker utility can also impact database performance caused by increased I/O contention when the utility is run while the database is available to applications.

IMS database administrators must balance the value IMS pointer checker utility analysis provides with the adverse affect it can have on running production applications. The questions that database administrators must ask are:

- What IMS pointer checker utility functions need to run?
- How often do they need to be run?
- How much impact will the utility functions have to production data?
- How can the impact to production data be reduced?

IMS pointer checker utilities can be performed on a cloned IMS database system. Running the IMS pointer checker utility on a cloned IMS system eliminates the production impact that would normally occur. Production IMS applications can maintain high availability and high performance while IMS pointer checker utility functions are performed on a cloned copy of the IMS system to validate data integrity. Performing IMS pointer checker utility functions on a cloned copy of an IMS system allows database administrators to run health check utilities frequently while preserving the availability and performance of running IMS applications.

IMS pointer checker utility can be offloaded to a cloned IMS system using the steps described in figure 1. IMS pointer checker utility is run in step number 7.

Alternatively, a cloned IMS system skeleton can be created using the steps described in figure 3. IMS pointer checker utility can be run on the refreshed database using the steps described in figure 2. IMS pointer checker utility is run in step number 8.

Creating Read-only IMS Environments

Many processing activities require read-only access to IMS databases. Creating IMS clone copies allows parallel read-only access to IMS data without impacting the performance and availability of production data. Offloading IMS read-only activities to IMS clone copies insulates critical production IMS applications from other read-only applications.

Some read-only processing activities, such as reporting and data warehouse loading, require the database to be in a known state and static while processing is performed. These types of processing activities limit production application availability during read-only processing activities. Production application availability can be maintained by creating an IMS clone copy and then performing read-only activities on the clone copy in parallel with production activities on the source IMS database.

Several examples below show how IMS performance can be maintained and availability improved by offloading and parallelizing read-only processing to an IMS clone.

Creating IMS Reporting Databases

Business reports are often generated after batch processing completes and before online processing begins. Online activities must wait for the reporting processes to complete before online processing can begin. Online processing can be extended by cloning IMS data after batch processing has completed; starting online activities on the production database, and then perform reporting processes on the cloned database copies. Performing reporting activities on cloned IMS data allow reporting to be done in parallel to online processing and on a static copy of the database.

IMS application reporting can be offloaded to a cloned IMS system using the steps described in figure 1. Reporting processes are run in step number 7. IMS databases can be refreshed in the cloned IMS system using the steps described in figure 2.

Alternatively, a cloned IMS system skeleton can be created using the steps described in figure 3 and application reporting can be offloaded to a refreshed IMS database using the steps described in figure 2. Application development and testing processes are run in step number 8.

IMS Staging for Data Warehouse Loading

IMS databases are the operational database for many enterprise applications. Many times data is extracted from operational IMS databases, transformed, and loaded into data warehouses (ETL) to support other enterprise applications. Extracting IMS data from production IMS databases can affect the availability and performance of production IMS systems. Extracting IMS data from a cloned IMS data allows parallel ETL processing on IMS data without adversely affecting production IMS applications and databases.

IMS data can be staged for data warehouse ETL procedures using the steps described in figure 1. IMS data extract procedures are run in step number 7. IMS databases can be refreshed in the cloned IMS system using the steps described in figure 2.

Alternatively, a cloned IMS system skeleton can be created using the steps described in figure 3. IMS data can be refreshed and staged for data warehouse ETL procedures using the steps described in figure 2. IMS data extract procedures are run in step number 8.

IMS Application and System Maintenance

Testing IMS Application Maintenance

Cloned IMS systems allow application developers to test new IMS application maintenance or release software easily using production IMS business data. All application testing can be performed by cloning the IMS system and running the new application maintenance or new release software on the cloned IMS system.

Performing IMS application maintenance or new application release testing using a cloned IMS system is done using the steps described in figure 1. IMS application testing is performed in step number 7. IMS databases can be refreshed in the cloned IMS system using the steps described in figure 2.

Alternatively, a cloned IMS system skeleton can be created using the steps described in figure 3. IMS database data can be refreshed to create application maintenance or new application release test environments using the steps described in figure 2. IMS application tests are run in step number 8.

Testing IMS System Maintenance

IMS system maintenance requires application validation testing and validation of maintenance procedures. This testing is done before the actual maintenance migration is performed and production use begins. Test IMS systems and database environments are needed to test maintenance acceptance and perform regression testing to ensure application compatibility.

IMS system maintenance testing can be performed on a cloned IMS system using the steps described in figure 1. New IMS maintenance can be applied and regression testing performed in step number 7.

Alternatively, a cloned IMS system skeleton can be created using the steps described in figure 3. New IMS maintenance can be applied before regression testing performed. IMS database data can be refreshed for applications to drive system maintenance testing using the steps described in figure 2. IMS application tests are run in step number 8.

Testing New IMS Release Migration Procedures

Testing new IMS release migration procedures using a cloned copy of an IMS system allows thorough release migration testing to be performed without affecting production IMS systems. New IMS release migration testing can be offloaded to a cloned IMS system using the steps described in figure 1. New release migration processes are run in step number 7.

Copying HALDB Databases

High Availability Large Database (HALDB) was created to provide high availability for large IMS databases using large data sets. HALDB databases can be partitioned such that they use multiple smaller data sets to support large databases. Using multiple smaller data sets to support a large database allows partitions to be copied, recovered, and reorganized faster and without affecting the whole database. HALDB database definitions and usage can happen without application changes.

Copying HALDB databases can be complex and can affect database availability. HALDB partition numbers are stored in the RECON master database record and in the database data sets. A partition cannot be opened if the partition ID in the data set does not match the partition ID in the RECON. Coordinating HALDB partition numbers complicates the copy process as HALDB partitions must be deleted and redefined in the target IMS and each HALDB redefinition must use the same partition number and gap sequence that exists on the source IMS.

Some HALDB copy processes use utilities to unload source database data and then load the data into a target database. These copy processes are slow and use significant amounts of host CPU and I/O resources. They operate on a single database at a time so all related databases need to be offline for time it takes to unload all of them. The database down time required for the copy processes has a significant affect on IMS application availability.

Other HALDB copy processes use image copy and recovery steps to copy HALDB databases. These processes run image copy on a source database and export HALDB

partition definitions. The database copy steps involved at the target IMS include: delete the target database definition from the RECONs, import the HALDB partition definition, register the image copy to the target RECONs and then perform a database recovery on the target database. Primary and secondary index and Indirect List Data Set (ILDS) rebuilds are performed after the recovery operation completes. Copying HALDB databases using an image copy methodology can be resource intensive, time consuming, and error prone.

RDR makes copying HALDB databases fast and easy. RDR uses storage-based fast-replication to copy the data instantly without using host CPU and I/O resources. It updates the target IMS RECON to reflect the same partition numbers as the source RECON. Primary and secondary indexes as well as the ILDS can be copied to eliminate index and ILDS rebuild time. RDR uses the steps shown in figure 2 to clone a HALDB database data and manage partition number and gap sequencing.