IN the late 1980s, IBM set its future direction for IMS database access with the incorporation of IMS/DBCTL. Introduced in IMS Version 3.1, DBCTL was designed to consolidate the access of IMS data from a single point: the DLISAS (Data Language Interface Subordinate Address Space) for full function IMS calls and the CTL (Control) region for fast path and DEDB (Data Entry) databases. This architecture would provide better reliability and recovery for DL/I applications as IBM moved forward with future plans in data-sharing and sysplex environments. For many CICS local DL/I customers, there was no immediate requirement to convert to or license DBCTL. Their DL/I applications could still operate without DBCTL in a local environment within CICS. But now, with Y2K issues and newer releases of IMS and CICS/TS, all local DL/I customers running CICS are faced with the daunting task of upgrading to DBCTL.

For strictly local DL/I sites, DBCTL introduces an entirely new subsystem environment. The tasks of installing, tuning and operating in this new environment require new skills. The systems programmer who is unfamiliar with IMS issues must make decisions about data access, recovery and batch conversions and implementations. Operations personnel must learn a new set of commands to control the environment: to stop and restart the IMS control region and perform similar functions for IMS batch message processing (BMP) regions. The news, however, is not all bad.

The easy part for current local DL/I applications is that source code requires no alteration to execute in the new environment. The calls to DL/I databases that were resolved in the CICS region using local DL/I are executed outside CICS via XMEM by the IMS control region. Though this mechanism may require additional overhead, benefits are realized when more than one CICS region is attached to DBCTL. Further, DBCTL provides several data recovery features, access to data entry databases (DEDBs) and BMP support that improves recovery and restart operations for those current applications. DL/I applications can also be reprogrammed to handle database scheduling failures; though it is doubtful that the source will be changed to implement this feature initially.

This article attempts to further explain the issues surrounding IMS/DBCTL migration and their performance implications. WHAT'S DRIVING ORGANIZATIONS TO IMS/DBCTL?

Y2K Support

Many organizations are still in the throes of Y2K evaluation and implementation. The effort can be very labor-intensive and time-consuming, not only to make application changes but also the required environmental upgrades. CICS sites running local DL/I applications are required to upgrade to IMS 5.1 or 6.1 for Y2K support. Installing, customizing and tuning IMS in an installation that previously did not have to deal with IMS issues is a difficult task to undertake. There are benefits to using DBCTL over local DL/I such as IMS logging, dispatching, storage management, latching, locking, and PSB scheduling, to name a few. However, supporting and maintaining DBCTL can be a challenge for those who are new to DBCTL. Early
planning, training of operations personnel, and a review with IBM may assist in reducing an organization’s exposure.

For current IMS 4.1 customers, there is a limited Y2K supported version available through a special RPQ release of IMS 4.1. However, the RPQ release for Year 2000 is entirely new and requires a complete reinstallation of IMS 4.1. Installing IMS/DBCTL 5.1 or above may be a better route.

Local DL/I Support Removed

CICS 4.1 was the last release to support local DL/I and is the lowest level of CICS that is Y2K-compliant. This means that customers using DL/I at the CICS 4.1 level must convert to IMS/DBCTL. In many cases, the performance hit of running with DBCTL vs. local DL/I is minimal, especially when it is necessary for multiple CICS regions on a single image to access many IMS databases. DBCTL can manage all the DL/I activity within a single DBCTL subsystem. This is likened to a CICS DOR (Data Owning Region) that services more than one CICS AOR/MRO (Application Owning Region/Multi Region Option).

IMS 5.1 and CICS Transaction Server 1.1 and above do not support DL/I calls within CICS. So the lowest level of Y2K-compliant software is CICS 4.1 with IMS/DBCTL 5.1. The table shown in Figure 1 presents release and compliance information. Figure 2 shows a typical CICS local DL/I implementation before DBCTL is implemented.

IMS/DBCTL also introduces at least three new regions: the control, or CTL, region; the database recovery, or DBRC, region; and the Data Language Interface subordinate address space, or DLISAS, region. The CTL region is started using an MVS start command. The IMS control region itself activates the other two. Operations staff familiar with communicating with CICS using an MVS modify command will have to adjust to a character interface similar to that used by JES2 or DB2. See Figure 3.

**MAJOR MIGRATION ISSUES TO CONSIDER**

**IRLM or PI?**

The internal lock resource manager (IRLM) provides database locking for the DBCTL environment. IRLM 2.1 for IMS allows up to 32 DBCTLs to share data. Before this release, you could share data across only two MVS IRLMs on the same two or different MVS images.

Though IRLM 1.5 is available, IMS 6.1 and later releases in which IRLM is used require IRLM 2.1 for sysplex data-sharing.

If you are a CICS installation that needs to share data only among CICS subsystems and BMP programs within the same MVS image, you do not need IRLM. Instead, you can use the program isolation (PI) facility (IRLM=N in member DFSPBxxx). PI will save a significant amount of CPU and reduce DL/I elapsed time.

**DBRC Setup**

Database recovery control (DBRC) controls logging within IMS. It also records recovery information in the recovery control (RECON) datasets, verifies that database utilities have the correct input, controls the recovery of registered databases and controls the data-sharing environment for purposes of serializing access to shared databases. This critical resource needs careful attention as your site plans for DBCTL.

All databases that require recovery need to be registered to DBRC. Non-BMP batch work applications can provide their own recovery by backing up the databases before execution. However, this method lengthens the recovery time if an abend occurs during the batch execution phase. DBRC simplifies the recovery process in such cases by allowing the batch work to restart from the last checkpoint without requiring the restoration of the database and subsequent restart of the abending batch step.

RECON datasets should be set up when you install DBRC itself. We advise you to include a minimum of three datasets: two that will be used; the second as a backup for the first; and the third as a spare. Ensure that these datasets remain on separate volumes and channel paths. We also advise you to make regular copies of databases using image copy utilities to

---

**Figure 1:** CICS and IMS/DBCTL Release Chart

<table>
<thead>
<tr>
<th>CICS Release</th>
<th>Y2K compliant</th>
<th>Indication</th>
<th>IMS/DBCTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3</td>
<td>No</td>
<td>Must convert to at least CICS 4.1 for DL/I support.</td>
<td>IMS 4.1</td>
</tr>
<tr>
<td>4.1</td>
<td>Yes</td>
<td>If Local DL/I must convert to at least IMS/DBCTL 5.1.</td>
<td>IMS 5.1 or above</td>
</tr>
<tr>
<td>TS 1.1</td>
<td>No</td>
<td>First release level that does not support local DL/I. DL/I applications must use DBCTL.</td>
<td>IMS 5.1+</td>
</tr>
<tr>
<td>TS 1.2</td>
<td>Yes</td>
<td>DL/I applications must use DBCTL</td>
<td>IMS 5.1+</td>
</tr>
</tbody>
</table>

**Figure 2:** Typical CICS Local DL/I Implementation Before DBCTL Implementation
ensure that your site has a comprehensive recovery strategy.

Logging and Recovery

IMS provides complete logging of all activity that occurs within its environment. Logging activity is critical to the successful recovery of data and provides guaranteed delivery of all “messages” within IMS. Such a generated level of logging granularity and its associated volume of information place a burden on systems resources. Take great care in evaluating the placement of these resources and how they will be used in a recovery situation.

The first consideration regarding IMS logging should be the write-ahead data set (WADS), which is used to temporarily hold IMS log records before they can be moved to the online log dataset (OLDS). Since WADS activity is usually very high, these files should reside on highly available, high-performance volumes. The WADS are involved in providing database integrity; therefore, you should use dual WADS rather than single (WADS=D). This way, DBCTL can recover committed works during emergency restart (in the event of a failure) — that is, data that has not yet migrated to the OLDS.

The DBCTL logging facility requires a minimum of three OLDSs to function. We recommend that you define more than three OLDSs in a production environment. In addition, define one large spare OLDS file as a dynamic allocation member and generate its definition in the IMS RESLIB for recovery purposes. This dynamic OLDS should not be started automatically during normal operation (control this through the DFSVSVMxx member). Use it only during emergencies when the other OLDSs are full but not yet archived. You can add the dynamic OLDSs via the IMS console /START command. When the OLDSs are full, DBCTL stops processing transactions and IMS appears to be stalled. This situation may concern operators who may mistakenly cancel or force IMS from the system, which could affect recovery. The dynamic OLDS can be used to relieve a stalled logging condition and allow processing to continue.

The location of the OLDSs is also important. If they reside on RAID devices, use only single OLDSs (RAID devices can provide better data redundancy and integrity). Otherwise, dual OLDSs are recommended for the production system.

CSA and Storage Usage

Evaluate the size and usage of current CSA below-the-line memory before installing DBCTL. Each DBCTL requires approximately 220K of common storage area (CSA) below the line. This limitation may affect current region sizes used by applications such as CICS. Attempt to reduce the number of DBCTls per MVS image by sharing multiple CICS AOR/MRO regions with the same DBCTL.

DBCTL Implementation

- **Database Definitions**: All the database definitions currently defined in the CICS region should be transferred to DBCTL. In general, database definitions should be generated with the RESIDENT option within DBCTL. Making all databases resident will prevent possible performance degradation when there is not enough database management block (DMB) pool space. The default is non-resident, mostly because older releases of IMS used less than 16MB of storage. The DMB pools are now above the 16MB line. If you specify non-resident and the DBM buffer pool is full, an open database may need to close and flush the DMB buffer pool and then open another database to fulfill the transactions request. In a CICS DL/I environment, the additional overhead may be unacceptable and will affect response times.

- **Trace Options**: The DBCTL trace feature provides the IMS debugger with the ability to see what is going on in a particular process. However, tracing is very expensive, especially in an active subsystem. Many of the traces are automatically enabled when IMS/DBCTL is installed. Several recommendations follow regarding this:
  - Turn the LATCh trace off since it could consume up to 10 percent of total CPU time used by IMS. You could use the Dispatcher (DISP) trace in test or QA systems to assist in debugging IMS problems. Do not turn on the DISP trace in the production system unless there is a specific need for it.
  - Limit use of the Monitor trace to 10 minutes. The impact on the...
Pool and Buffer Tuning: Pooling and buffering in IMS moves data into memory in stages. The utilization and size of each pool and buffer may vary. Access method pools should be large enough to reduce the number of I/Os to databases. Index subpools should be separated from data subpools. Usually, index subpools range from 512 bytes to 2KB. Data subpools should range from 4KB to 32KB. The number of pools may vary based on the requirements of the applications running. A rule of thumb: Index subpool hit ratios should be greater than 85 percent and the data subpool ratio greater than 70 percent. Certainly no environment is the same, but these ratios are widely accepted.

IMS System Checkpointing: IMS offers the highest level of integrity and recovery on the S/390 platform. No transactions or “in-flight” work is lost because of IMS logging and system checkpointing. Backing out transactions and recovering from application or system outages is a major expertise of IMS.

IMS takes system checkpoints at specified intervals based on the CPLOG value parameter. During a checkpoint, IMS seems to stall briefly because of the amount of data being written to the IMS log files.

PSB Workpools: The PSB workpool (PSBW) is used to pass DL/I data between the DLISAS address space and CICS. When the PSBW pool is full, further PSB scheduling is curtailed. When this occurs, the thread is suspended on a “block-loader wait” until the pool space is available. The result is a drastic slowdown in CICS transaction throughput. To estimate a reasonable value, find the concurrency rate for each PSB, then multiply that number by the PSB size (shown in the PSB generation). The optimum size for the PSBW pool should be initially 10MB or more.

Then you can adjust the value downward to the PSBW high-water mark (plus 20 percent for fragmentation). The most accurate method for calculating this value is to issue the /DIS POOL command just before shutting down IMS. The high-water mark for all pools will be displayed. Or, you can run the DB Monitor batch reports.

Dispatching Priorities: In a busy environment, it is important to place DBCTL address spaces in the most appropriate dispatching priority. We suggest that you place the server address spaces at a higher priority than the requester for services. If IRLM is used it should have the highest priority, followed by the CTL, DBRC and DLISAS regions. CICS dispatching priority should be one notch below these. Since DL/I services are performed under cross-memory mode in most cases, the dispatching priority is dependent heavily on the caller’s TCB priority. As a result, DBCTL CPU utilization will be very low since the CPU time will be charged to CICS or BMP.

BMP Conversions: There are several advantages to converting a normal batch CICS update job to a BMP. BMPs provide support for concurrent update of databases while DBCTL and CICS are active. Application syncpoints within a BMP provide for recovery and restart from the last syncpoint without having to recover all the databases that have been updated during the run. JCL is also simplified since the databases are defined only in DBCTL and not in the BMP’s JCL. Again, you don’t have to change the
source code to take advantage of these new facilities; however, you do need to change the production JCL.

Operational Differences

Operations personnel are faced with numerous challenges related to the operation of DBCTL. It appears to be a full-blown IMS environment with a control region, a DLISAS region, a DBRC region and a possible IRLM region. Operations will need to know how to operate in the new environment to start and stop the control region, open and close databases, and stop and restart batch BMP regions. Early planning and training of operations are key elements.

Starting and stopping regions and batch work is very different if BMPs are in use. Operations must stop and perform restarts for BMPs using IMS commands rather than MVS commands. Each BMP can be stopped using the /STOP REGION command once the region ID is obtained from a /DISPLAY ACT command. In other words, terminating a BMP is a two-step process.

WHAT ARE THE BENEFITS RECEIVED?

DBCTL reduces the recovery time for failed CICS transactions and batch jobs. Oftentimes, batch recovery in the CICS environment requires the restoration of a database and the re-execution of the batch job stream. BMP regions with sync-point-restart can eliminate not only the data recovery step, but also allow the application to restart in the middle of a processing step.

CONCLUSION

Certainly, there are many more issues in managing the performance and availability of a DBCTL subsystem, including DBRC, database design, data sharing, and workload balancing. However, we hope this discussion provides you with a good starting point from which to address the requirements of performing the conversion from DL/I to DBCTL.

Several resources from IBM may also be helpful to you, such as “DBCTL Migration and Implementation,” a videotape made by Dave Viguers and Steve Zemblowski, which reviews the process, as well as IMS/ESA Version 6 Release Planning Guide (GC26-8744) and IMS/ESA Version 5 Release Planning Guide (GC26-8031).

Loc Tran is a senior development specialist at Landmark Systems Corporation. He has more than 20 years of experience as an IBM IMS and MVS systems programmer. He can be contacted via email at Loc.tran@landmark.com.

Mark Hedges is a product manager at Landmark Systems Corporation in Vienna, VA. He can be contacted via email at mhedges@landmark.com.

© 1999 Technical Enterprises, Inc. For reprints of this document contact editor@naspa.net.