



Sparkassen Informatik Utilizes Data Sharing to Guarantee High Availability to Its Customers

Marlies Baute

Sparkassen Informatik |

Sparkassen Informatik is the largest IT service organization for German savings banks, supporting more than 250 savings banks and approximately 9000 branch offices in Baden-Württemberg, Hessen, Nordrhein-Westfalen and Rheinland-Pfalz.

Sparkassen Informatik was formed in 2001 when IT service providers Sparkassen Informatik Baden-Württemberg, Sparkassen-Informatik-Systeme West (SIS West), and Informatik Kooperation merged. On October 1, 2003, Sparkassen-Informatik-Services West (swest) also consolidated into Sparkassen Informatik.

Headquartered in Frankfurt, Sparkassen Informatik employs approximately 2700 people. Additional area-wide locations provide service support for savings banks across west and southwest Germany, ensuring quality service and premium technical and business support.

While the portfolio of Sparkassen Informatik covers the entire spectrum of information technology, the company's primary focus is the support of business strategy of the connected savings banks. Technology and applications are designed to achieve these customer goals.

Our Solution with IMS DB/DC and Parallel-Sysplex

Savings banks in Rheinland and Rheinland-Pfalz use IMS Version 8 to run their transactions. Each day, we run approximately 40 million transactions.

To manage this high volume of transactions, we have installed six IMST[™] systems. One of our IMS systems is an ATM routing system with 5300 automated teller machines, 6200 self-service statement printers, and 1700 information terminals. The other five run with more than 95 000 terminals and 33 000 printers. At peak times, we run 1800 transactions/second.

Most of our databases are Data Entry Databases (DEDBs). Some of them are shared Virtual Storage Option (VSO) DEDBs because they are extremely fast and guarantee high performance. To improve availability for update access, all DEDBs use multiple area data sets (MADS). We are also using full-function databases and DB2[®] tables.

Providing Internet banking services 24 hours a day, 7 days a week is a demand of our customers. We conduct Internet banking using IMS Connect. To provide continuous service, we converted our IMS systems into two-way data-sharing systems with shared queues. This system also enables us to provide continuous ATM operation.

Converting our IMS systems to two-way data-sharing systems was a challenge. To enable our applications and databases to run in a sysplex environment, we had to:

Change and test applications with SCHEDTYP=PARALLEL. SCHEDTYP=SERIAL was the default for macro APPLCTN.

Change our Main Storage Databases (MSDBs) to shared VSO DEDBs.

Change the databases to share level 3. Some databases had to be reorganized or redesigned to make them uninterrupted.

continued on next page . . .

In This Issue:

Letter to Customers	3
IMS Connector for Java	4
IMS Service Corner	6
IMS Performance	8
IMS Profiles	9
IMS Tools	10
IMS OTMA	12
Upcoming Events	13
Redbooks	14
Hot Websites	15
Editorial	16
Trademarks	16



... continued from previous page

Other challenges we overcame include applications that had coded the IMSID, applications using supervisor calls (SVC) to get a serial number, and real storage tables.

To migrate our systems to two-way data-sharing with shared queues, we had to:

Change the IMS locking from PI to IRLM LOCAL, then to IRLM GLOBAL.

Implement the MVS™ Logger, which resulted in either a performance loss during Common Queue Server (CQS) structure checkpoints or a very large space consumption of log data.

Activate VTAM® Generic Resources for advanced program-to-program communications (APPC).

Implement Sysplex Distributor and virtual IP address for IMS Connect.

Change IMS dependent jobs to use the IMS group name instead of the IMSID.

Update data sets with new naming conventions.

Implement scheduler environments to make job distribution easier.

Establish Fast Database Recovery (FDBR) to monitor IMS, detect failures, and recover any database resources that are locked by the failed IMS, making them available for other IMSs.

Implement new functions for system automation (SAFOS 2.1) to change automation and surveillance regarding queues.

Change monitoring in regard to global locking conflicts and different queue types like serial transaction queues.

Install the Performance Analyzer to get reports on sysplex activity. The easy-to-use Performance Analyzer merges logs internally.

Install DELTA PLEX and DELTA IMS Virtual Terminal. These tools enable us to create and modify all parameters in the IMSplex. They can also send commands to all IMSplex members.

Install the IBM® IMS Queue Control Facility (QCF) to replace the IMS Message Requeuer. We use QCF to delete old queue data and to get rid of messages dedicated to the cold queue.

Image copy all database types with the Image Copy 2 utility. With two-way data sharing, full-function databases (strictly VSAM KSDSs) need Image Copy 2. Without it, you have to make sure that the database is active only in one IMS member in the IMSplex.

With implementation of the second IMS (a clone of the first one) in the IMSplex, we expanded automation and surveillance for Intersystem Communication (ISC). If there is a connection loss, automation attempts to recover the connection and sends an alert. Automation is not affected by which IMSplex member had established the connection. We had to make sure that alerts were sent only when both members did not have a connection. With this connection, we were able to run our ISC connections without interruption.

Dependencies in some applications resulted in not being able to run the IMSplex members in parallel. In case of an outage, we had to make sure that the regions and dependent systems had to be started on the other IMSplex member. We achieved this by creating more groups in system automation to keep the interruption time frame as short as possible.

After this migration effort, we are now glad to guarantee the high availability of our IMS systems for the benefit of our customers. Functionality for sysplex environments introduced with IMS Version 7 and Version 8, including the ability to upgrade the RECON data sets online and the auto alter for shared VSO and shared queues, has been especially helpful.

Our current IMS configuration, operating with data sharing in a Parallel Sysplex® environment, is essential in meeting future challenges at Sparkassen Informatik and their demanding customers. □

Marlies Baute



IMS and the Product Introduction Center

Over the many years of its life, IMS has become a vital part of our everyday online systems, with millions of people across the world relying on IMS technology to process their data, handle their transactions, or deliver cash on demand using automated kiosks or ATMs. When a new version of IMS is released, it is obviously very important that the existing systems are tested prior to their migration to ensure that no major bugs are introduced that would cause systems to fail or perform incorrectly. In addition to this testing, there is also the need to test the new IMS features and facilities in each release, which allow users to better manage, integrate, and deploy their systems.

The IBM Global Product Introduction Center (PIC) is a group whose mission is to speed the deployment of new products to customers. This deployment is achieved by:

Working with the IBM product development teams to understand the planned functions and delivery schedules of new product releases.

Organizing managed beta programs, sometimes referred to as Early Support Programs (ESPs) or Quality Partnership Programs (QPPs), with key customers to ensure the products are used and tested prior to their release, or to use the IBM term, General Availability.

There can be many different parties involved in a beta program, and the needs of each must be balanced to ensure success:

Customers would like early access to the code to ensure there are no migration issues that will affect them, and also to test new features to see if they can be used in their environment. This can lead to early adoption of new technology and competitive advantage, or reduced costs as new features allow greater efficiencies.

Development teams would like their code tested to ensure that it performs correctly and is as bug-free as possible. Testing in diverse customer environments and by users who have not been involved in the development process is a key part of discovering where the real issues will be when the version is released. Developers also appreciate customers' ideas for new product features to be considered for future versions.

Marketing teams would like customers to become references for the new versions as they become available, to generate positive publicity and potentially help to convince new users to buy the product.

IBM technical teams, both in sales and support, would like to have early hands-on access to the new product versions and to receive education on its use and features.

The PIC has teams in North America, EMEA (Europe, Middle East, and Asia) and Asia Pacific. These teams work with many development groups within IBM to identify suitable beta customers and test their products in areas such as:

DB2 Information Management Systems
WebSphere® portal and integration technologies
Lotus® Workplace, Notes®, and Domino®
Tivoli® systems management

Some of these programs can be relatively small and self-contained, running for three months with one or two customers just to test key functionality. However, programs tend to be longer on the key mainframe systems to ensure a complete and rigorous approach to testing; recent examples would be the IMS Version 8 QPP, and the DB2 Universal Database™ for z/OS® Version 8 programs.

The program manager for IMS in the PIC is Alison Coughtrie, based in Hursley, United Kingdom. Alison has recently completed work on the IMS Version 8 QPP and is now ramping up her efforts on the IMS Version 9 QPP, which will see activity throughout 2004. For Alison, one of the key benefits in the QPP approach is the relationships that grow between the customers and the development teams over the course of the program.

Alison says, "An IMS QPP can last 2 years from the initial planning to the completion stage. We can schedule education sessions, where customers are taken to the labs to meet the developers and learn about the new features. Customers get early access to the code and can give feedback on how they are progressing with their testing and tell us what issues they have encountered. As QPP participants, customers have frequent contact with the development teams and most issues are fixed very quickly. At the end of the program, we collect feedback from the customers on the program itself and also on the product and feed this information back to the labs, which helps us improve our products and helps the customers in future. We often find customers coming back to us again and again asking to be involved in the next QPP."

Key IMS customers seem to agree. GAD, based in Munster, Germany, wrote about their QPP experiences in the IMS Newsletter in Winter 2002. They described the overall experience as very positive, and the support from the PIC as "in a word, perfect." High praise indeed from a customer who demands very high standards from their products and services.

If you would like further information on the IMS QPP or on beta programs for other IBM products, please contact me or Alison Coughtrie alison_coughtrie@uk.ibm.com.

Adam Jefferson
EMEA PIC Manager
Hursley Park, UK
ajefferson@uk.ibm.com



IMS Connector for Java Two-Phase Commit Support

Protecting and maintaining the integrity of critical business resources is a key issue for most enterprises. When you make changes to your protected resources, you want to guarantee that the changes are made correctly. For example, as a bank customer you want to transfer money from your savings account to your checking account. You want to be sure that when the money is deducted from your savings account, it is added to your checking account simultaneously. You would not want this transaction to be completed only partially, with the money deducted from your savings account but not added to your checking account.

In this example, several smaller transactions are required to complete one overall transaction. If there is a problem with one of these smaller transactions, you would not want to commit the overall transaction (such as transferring money). Instead, you would want to roll back every step of the transaction so that none of the smaller transactions is committed.

This can be achieved by coordinated transaction process and two-phase commit processing.

What is Two-Phase Commit?

Two-phase commit is a two-step process by which recoverable resources (for example, databases) are coordinated to commit by a transaction coordinator. A transaction coordinator provides services to support and manage transactions; it communicates closely with one or more resource managers (for example, database managers) to coordinate changes in a transaction.

During the first phase, each resource manager is polled by the transaction coordinator to ensure that it is ready to commit; for example, the database manager is asked if it can make the changes permanently to the database. If all resource managers respond positively, the coordinator makes the decision to commit. After the decision is made, all the changes will be committed, even if there is a failure. If one of the resource managers responds negatively, the coordinator will decide to rollback, in which case no changes should be made to the resources. In the second phase, the coordinator instructs the resource managers to commit or rollback, depending on the decision from the first phase.

How IMS Connector for Java Two-Phase Commit Works

IMS Connector for Java supports global transaction management and two-phase commit processing for both distributed and z/OS environments. Using this support, you

can build a J2EE application to access IMS transactions and other resources (for example, DB2 on distributed platforms) in a single unit of work. All changes within a transaction, therefore, are either fully completed or fully rolled back. This support enables your application to access IMS transactions and data in a coordinated manner and ensures the integrity of the data in IMS.

IMS Connector for Java is designed to work with the Java transaction manager in WebSphere Application Server (WAS), the Resource Recovery Services (RRS) of z/OS, and IMS Connect, to make consistent changes to IMS and other protected resources.

Figure 1 uses the money transfer example to illustrate how your application uses IMS Connector for Java and IMS Connect to participate in two-phase commit processing:

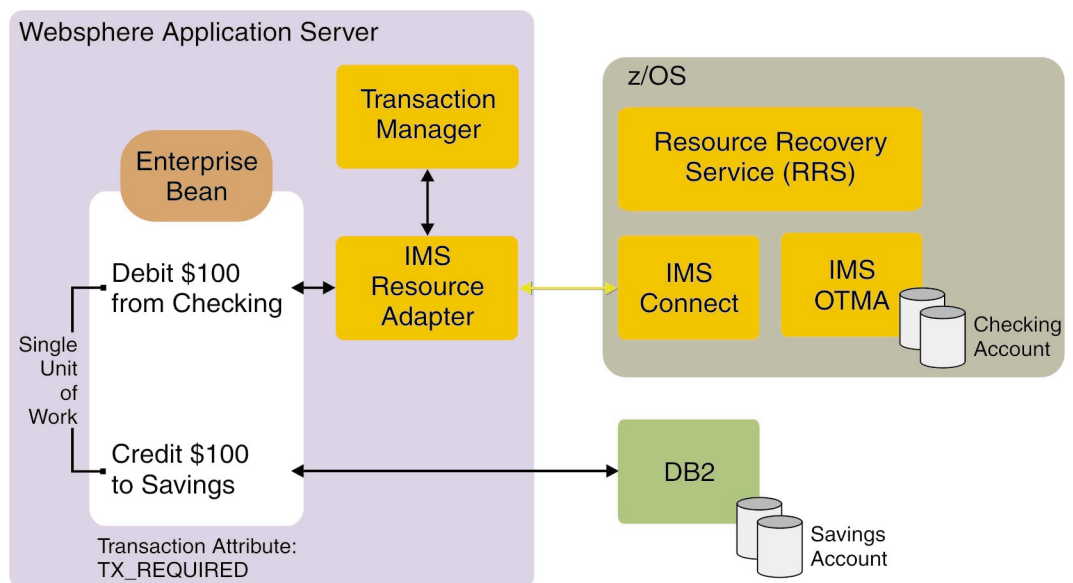


figure 1

Your objective is to transfer \$100 from your savings account (stored in an IMS database on the z/OS platform) to your checking account (stored in a DB2 database on the Windows platform). Figure 1 demonstrates how a J2EE application with an Enterprise JavaBean (EJB) transfers the money by invoking the transactions to:

- Run an IMS transaction using IMS Connector for Java and IMS Connect to debit \$100 from the checking account.
- Run an SQL query to update DB2 to credit \$100 to the saving account.

This EJB is defined with TX_REQUIRED transaction attribute to tell the Application Server that the two transactions are running as a single unit of work. If both transactions complete successfully, the two transactions are committed by the transaction coordinator (or transaction manager) in the J2EE server (for example, WebSphere Application Server). The

transaction manager works with the IMS resource adapter to coordinate the changes in IMS, and with DB2 to coordinate the changes using two-phase commit.

IMS Connector for Java supports two-phase commit in both z/OS and distributed environments. If your application is running on WebSphere Application Server for z/OS, and your IMS Connect, IMS, and RRS are all in the same z/OS image, you can use the Local Option communication to execute your application with two-phase commit processing. In this environment, Local Option uses z/OS Program Calls to communicate with IMS Connect, and transaction processing is coordinated by RRS and WebSphere Application Server for z/OS.

If your application is running on WAS, and that server is running either on distributed platforms, on the same z/OS on which IMS Connect is running, or on a different z/OS, you can use TCP/IP communication. This scenario requires that RRS, IMS Connect, and IMS run on the same z/OS system to enable two-phase commit.

IMS Connector for Java interacts with the WAS transaction manager using the industry standard X/Open XA protocol to manage the two-phase commit processing. IMS Connect interacts with RRS to coordinate changes with IMS. This new XA TCP/IP two-phase commit support is available with IMS Connector for Java Version 2 Release 1.

How You Can Use Two-Phase Commit in Your Application

Two-phase commit support in IMS Connector for Java fits seamlessly with WebSphere Studio Application Developer Integration Edition (WebSphere Studio). The tooling is based on the Web Services model, which provides a higher level of code reuse. To use WebSphere Studio to build a two-phase commit application, you must:

1. Create the enterprise service. The enterprise service is a scalable, reusable web component built to exploit function provided by a J2EE server. WebSphere Studio provides wizards to generate an enterprise service. Using WebSphere Studio, you avoid the manual coding of a simple Java application. Instead, WebSphere Studio imports a COBOL file to mimic the input and output records provided to and received from an IMS transaction. WebSphere Studio generates WSDL files and proxies to incorporate applications into EJBs. WebSphere Studio also generates the EJBs to encompass the Java application, the EAR file, the EJB packaging, XML documents, as well as the overriding WSDL files for future construction of EJB proxies. Several examples of generating enterprise services for IMS transactions are shipped with WebSphere Studio.

2. Generate an EJB proxy. WebSphere Studio's automation wizards enable you to generate an EJB proxy from the enterprise service WSDL files. An EJB proxy provides a simple Java class and methods to allow a J2EE client, EJB, or web page to access the service hosted by a J2EE server.

3. Generate container-managed and component-managed EJBs. After the enterprise service and its proxy are completed, application EJBs can be coded to bundle multiple service calls into a single transactional unit of work, or a two-phase commit transaction. The transaction scope can be managed by the container or bean (component). You can easily identify the transaction management as either Container or Bean by modifying a transaction type field in the EJB description XML. For the container-managed EJB, you can assign transactional attributes to the business methods of an EJB to demarcate

transactional boundaries. These attribute values instruct the J2EE server to have that business method begin, join, or seclude itself from a transaction. The alternate and more manual means of transaction demarcation, component-managed or bean-managed EJBs, can also be used. The application programmer would have to code the transactional boundaries without the aid of the J2EE server.

The tooling is based on the Web Services model, which provides a higher level of reuse.

WebSphere Studio allows you to easily code the EJB methods, promote them to become business methods (rather than internal EJB methods), and assign a transaction attribute to those EJB methods built within a container-managed EJB. WebSphere Studio also supplies deployment code, access beans, and the means for generating web pages based on the business methods of the EJB.

After development of the J2EE application, you can export the EAR file, and deploy it to your remote test or production server. Before using the application, use the IMS Connect command, VIEWHWS, to verify that the application has been enabled to communicate with RRS. A field should be displayed indicating RRS=Y on the console. You should also ensure that IMS has been set up to communicate with RRS, and that the appropriate transaction has been started and has an abundant amount of MPP regions to run without contention. After verifying this, you can then begin to use your two-phase commit application.

continued on page 7 . . .

Improving Service to Simplify Maintenance

Maintaining your IMS systems can be a challenge in today's complex environments. In IMS Service, we developed the following guidelines to help you develop a maintenance strategy. These guidelines include many of the latest developments in the service arena to help simplify the service process. You can also find this information in the IMS PSP (Preventative Service Planning) Buckets under IMS710 HMK7700/GA2, and IMS810 HMK8800/GA, where you can always obtain the most current version of the IMS maintenance recommendation.

General IMS Maintenance Recommendation for an IMS Production System

Start with an IMS production system base implementation service level:

1. Install service levels 6 months behind the current ESO/CBPDO level for planned migration
2. Install all HIPER PTFs fixes 3 months behind the planned migration date
3. Resolve PEs
4. Conduct a 3-month test cycle prior to production implementation

Ongoing IMS Production System Service Level Recommendation

IMS maintenance continues to be distributed during the 3-month test cycle. We recommend that you include the following in the initial IMS production base system implementation:

- Install fully tested fixes for significant software problems; by significant, we mean that serious consequences might be encountered without the application of the fix.
- Install fully tested HIPER SYSMODs that are directly applicable to the specific IMS environment and are significant.
- Review the IMS PSP bucket UPGRADE for the IMS release level and SUBSET FMIDs; important IMS product information is continually added to these buckets. You might be prompted to take a variety of actions to keep your IMS system in proper working order.

The following is an example of how to implement the recommended maintenance. Refer also to informational APAR I108928 (for IMS Versions 7 and below), and APAR I1013024 (for IMS Version 8):

1. Obtain current service

Use your current service delivery method, or try ShopzSeries. ShopzSeries is IBM's web-based productivity tool, which simplifies ordering of zSeries® software products, product upgrades, and system maintenance. See www14.software.ibm.com/webapp/ShopzSeries/ShopzSeries.jsp.

2. Obtain and receive current Enhanced HOLDDATA

See service.boulder.ibm.com/390holddata.html for the most current information regarding Enhanced HOLDDATA.

3. Use SMP/E to install the service

Select RSUyymm SOURCEIDs that are created by CST (Consolidated Service Test). See www.ibm.com/servers/eserver/zseries/zos/servicetst/ for additional information regarding CST. Then resolve both PEs and System HOLDS.

4. Obtain and receive current Enhanced HOLDDATA again

Run SMP/E REPORT ERRSYSMODS to identify HIPER/PE exposure. Obtain applicable SYSMODs, and process SYSMODs that are applicable to your environment. Use the IMS Support web site or PSP Buckets for APAR descriptions. If you find that some PE SYSMODs are already on the system, you can:

- Remove the PTF in error if it is not already accepted.
- Leave the PTF in place if the reported PE symptom is not significant.
- Establish operator procedures to restrict access to reported areas of exposure.
- Apply a corrective APAR/PTF fix, if available.
- Request a FIXTEST from software support for the reported problem if the APAR is still open.
- Request a USERMOD code bypass for the reported problem from software support.

5. Review the most current IMS PSP Buckets for new service information

6. Test the new maintenance level

Repeat steps 4, 5, and 6 in an ongoing basis to remain current.



Caution: SYSMODs in APPLY-only status could be regressed by SYSGEN. There are two methods of preventing this. Either ACCEPT all PTFs in APPLY-only status before SYSGEN, or, after SYSGEN, reprocess the APPLY-only SYSMODs.

The following is an example of the SMP/E statements that can be used to supply a list of SYSMODs in APPLY-only status:

```
SET BOUNDARY (targetzone) .
LIST APAR PTF USERMOD NOACCEPT NOSUP .
```

The following is an example of the SMP/E statements that can be used to reprocess SYSMODs in APPLY-only status:

```
APPLY REDO NOJCLIN SELECT (
    xxxxxxxx
    xxxxxxxx
    xxxxxxxx
)
```

In this example, xxxxxxxx explicitly lists each SYSMOD in APPLY-only status. NOJCLIN should only be specified for REDO processing. Some SYSMODs in APPLY-only status might have System HOLDS for SYSGEN that require special handling. Follow the instructions in the HOLD statements for those SYSMODs.

Improvements in the service arena continue to simplify the maintenance process for IMS. As you change your procedures to take advantage of these improvements, we will be happy to assist you in your maintenance efforts. Simply open a PMR and send it to the IMS support team if you have any questions or if you encounter any problems. □

Jeff Maddix

IMS Service

John Butterweck

IMS Service

... continued from page 5

Two-Phase Commit Performance Tips

To maximize the performance of a two-phase commit application, consider the following:

A transaction that is within a two-phase commit application will continue to use an MPP region for the duration of the entire two-phase commit transaction. If a number of IMS transactions are performed within a two-phase commit transaction, at least that many MPP regions must be available to avoid hanging the two-phase commit application. For performance reasons, it is best to have as many MPP regions as possible running to ensure that two-phase commit applications do not contend for a region.

Similarly, you should avoid having an excessive number of database interactions performed in one two-phase commit transaction.

If multiple IMS transactions are used within a two-phase commit transaction, they could possibly contend or lock in an attempt to update or modify the same data. To avoid this, it's best to write an application that will prevent a user from accessing duplicate entries within the same two-phase commit operation. Also consider configuring your IRLM or PI locking manager to use a block size that is as small as the smallest entry to that database. Larger block sizes might have two transactions contending for entries that might not even be the same and yet reside close to one another on the hard disk.

For more information about two-phase commit, including sample applications, see *IMS Connector for Java Guide and Reference*, SC27-1559. □

Haley Fung

IMS Developer

James Polo

IMS Developer

XCF Tuning for IMS Running in a Parallel Sysplex Environment

The z/OS Cross-System Coupling Facility (XCF) signaling service has been widely used in IMS to send and receive messages between XCF members. To ensure the best XCF response time and message throughput capacity in a Parallel Sysplex environment, XCF tuning for IMS and IMS Connect might be needed. This article describes XCF tuning tips for the transport class definitions, buffer definitions, and signaling paths.

As XCF messages are generated, they are assigned to a transport class based on group name, message size, or both. The messages are copied into a signal buffer from the XCF buffer pool. The messages are sent over outbound paths (PATHOUT), defined for the appropriate transport class. Messages from other systems are received by inbound paths (PATHIN). The key to ensuring good performance for the XCF signaling service is to provide sufficient signaling resources, namely message buffers, message buffer space, and signal paths, and to control access to those resources with the transport class definitions.

Tips for Transport Class Definitions

In most cases, it is sufficient to define the transport class based on message size. You can do this by specifying UNDESIG on the GROUP parameter of the CLASSDEF statement in the COUPLExx parmlib member. This ensures that the XCF messages are assigned to the transport class based solely on message size.

There might be times when you want a separate transport class for a specific group for problem diagnosis purposes. After you diagnose the problem, you can reassign the messages to a transport class based on the length of the messages.

You can use the Resource Measurement Facility (RMF) XCF report to determine how well the messages fit between different transport classes.

Keep the number of transport classes small to efficiently pool the resources.

Tips for Message Buffer Definitions

The XCF message buffer size is determined by the CLASSLEN parameter on the CLASSDEF statement in the COUPLExx parmlib member. Set the CLASSLEN value with the size of the most frequent message expected in this transport class. For example, one transport class, DEFSSMALL, can be used for messages less than or equal to 956 bytes with the following definition: CLASSDEF CLASS(DEFSSMALL) CLASSLEN(956) GROUP(UNDESIG). The transport class DEFAULT can be used for messages greater than 956 bytes with the definition CLASSDEF CLASS(DEFAULT) CLASSLEN(16316) GROUP(UNDESIG).

You can issue the XCF command D XCF,CLASSDEF to list the CLASSLEN specified in the parmlib member.

You can review the RMF XCF report to see the value of %SML, %FIT, and %BIG for messages that fit the CLASSLEN value.

It is important to use the MAXMSG parameter to select an appropriate upper limit for the amount of virtual storage to be allocated to the XCF message buffer pool. The message buffer pool is virtual storage that XCF uses to store the message buffers that are being processed, sent, or received.

For outbound messages to a particular system, if the sum of the buffer storage for CLASSDEF and the outbound paths is insufficient, the XCF signal is rejected. This is reported on the RMF XCF report as REQ REJECT for OUTBOUND requests. You can increase MAXMSG on either CLASSDEF or the PATHOUT definition to resolve the problem.

For inbound messages from a particular system, if the storage for the inbound paths is insufficient, the XCF signal is delayed. This is reported on the RMF XCF report as REQ REJECT for INBOUND requests. You can increase MAXMSG on the PATHIN definition to resolve the problem.

You can check the RMF XCF PATH STATISTICS report to see the BUSY counts. High BUSY counts can be caused by an insufficient number of paths or a lack of inbound space.

Local messages are signals within the same image. Code MAXMSG on the LOCALMSG statement to get additional message buffer storage for local messages.

Tips for Signaling Paths—CTC Links or Coupling Facility Structures

For availability, a minimum of two physical paths must be provided between any two systems. Only for environments with very high rates of XCF traffic would additional paths be required.

You can tune signaling paths by altering the number, type, or both, of paths. To review the XCF path configuration, use the RMF XCF PATH STATISTICS report.

To determine if the “No Paths” condition exists, look at the RMF XCF Usage by System report. ALL PATHS UNAVAIL should be low or 0. In many cases, the “No Paths” problem is caused by an error in the path definition; in other cases, there might be a problem with the physical path.

For Coupling Facility (CF) structures, an insufficient number of PATHOUT links could result in:

- An increase in the AVG Q LENGTH
- BUSY counts being high relative to the AVAIL counts as shown in the RMF XCF PATH STATISTICS report.

If you use CF structures as XCF paths, make sure the structure size is adequate. If the structure is too small, you will see an increase in the number of REQ REJECT and AVG Q LENGTH, and these events will definitely affect response time. Review the CF CPU utilization as well.

For CTC links, the XCF command XCF,PI,DEVICE=ALL,STATUS=WORKING can be issued to display the path response time. See the MXFER TIME column for the mean transfer time in microseconds. The AVG Q LEN on the RMF XCF PATH STATISTICS report is not a good indicator of the transfer time. Review the channel utilization for the CTCs, as reported on an RMF Channel Activity report, as well.

In high volume XCF environments, be sure to have a CTC configuration plan that distributes the CTC I/O load across all available SAPs.

Special thanks to Sinmei DeGrange for sharing her XCF tips for running IMS Connect performance tests.

For more information on the reports that RMF produces for tuning a sysplex, see *z/OS MVS Setting Up a Sysplex and RMF User's Guide*.

For more information about XCF tuning, see Washington Systems Center Flash, *Parallel Sysplex Performance: XCF Performance Considerations (Version 2)*. □

Jack Yuan

IMS Developer



Shahin Mohammadi-Rashedi is just starting her 22nd year with IBM. She began her career in Kingston, New York, and came to Silicon Valley Laboratory—and IMS—in 1991. Shahin is part of the IMS Tools Support team. She works with others to create and support tools that enable IMS Developers and Service providers to do their jobs better. “I try to stay ahead of the game,” she said, to anticipate how to best support her customers. Although Shahin’s primary focus is on IMS Tools support, she also provides support throughout the entire lab when necessary. Her current focus is on IMS’s ATC—Application Testing Collection. With ATC, Shahin and her Tools Support team contribute to an enhanced test effort that ensures that every line of new and changed IMS code is fully tested. Shahin stated that she has “been very fortunate” to work within the IMS organization. A self-confessed “outdoor person,” Shahin appreciates the beauty of the lab’s location and enjoys walking on our nearby hillsides. □

Shahin Mohammadi-Rashedi

Greg Vance

Greg Vance has been with IBM—and IMS—for 16 years. Greg joined IBM after completing his undergraduate education at UCLA. He completed his post-graduate education at Santa Clara University while working for IBM. Greg started his IMS career in Test, recreating problems and solving them by asking himself, “how did this happen in the first place?” Greg has been the IMS test coordinator for several IMS releases; his first test lead experience was on Remote Site Recovery (RSR). After honing his skills there, he looked for new challenges and made key contributions to the Database Recovery Facility (DRF) team. Now working on future enhancements with the DBRC development team, Greg says that he enjoys working with the people in IMS, and also working with customers when he gets the chance. He still enjoys tackling test problems when he can, and says, “I like to help solve problems, even when it’s not my problem.” When he’s not in problem-solving mode at IBM, Greg can be found camping and water skiing with his family (he has two kids), and contributing his audio and sound skills at his church. □



What's New in IMS Tools?

It's that time of year again - time to take a few minutes to reflect on 2003 accomplishments and time to start executing 2004 plans. The IMS Tools group is doing the same thing - we've had a very successful 2003 and are very excited about our 2004 plans.

Did you notice early in 2003 that our name changed? In some places, you might still see both names as we work towards updating the wealth of IMS Tools material that exists. The name change was driven by our need to support Software Group's cross-product rebranding of **DB2 Information Management Software**. So, **DB2 and IMS Tools** is the new name for the product group formerly called **Data Management Tools for DB2 and IMS**. Personally I like it—no more questions about what's in Data Management. Now everyone knows what and who we are.

Not only did we change our name in 2003, in December we changed our tool categories and the mapping of tools within those categories. We've grouped our IMS tools into six functional areas to match your database and system requirements:

administration tools to maximize database availability
utilities to handle a wide variety of database tasks automatically

performance management tools to keep systems and databases running smoothly

recovery management to assist with unique backup/recovery situations

replication management to improve the availability of copied data

application management tools to help you create and operate application subsystems

In addition, there are tools specially designed to help you with **IMS TM management**. You can find more information on the categories and tools within the categories at our website: www.ibm.com/software/data/db2imstools.

Besides these "housekeeping" changes, we've had lots of individual IMS Tools changes in 2003. We had four tools announcements - April 29, June 24, Sept 9, and Dec 2. With these announcements, we've added five tools to the IMS Tools portfolio, one of which is available only through Early Support Program participation, eight enhanced tools, and one new tool being offered as a PRPQ. Like I said, we've been busy.

Here's a recap of 2003's activities:

IMS Multi-Dialog Manager for z/OS, V1.1 (5697-H91) -

This new tool enables your IMS users to more quickly change between IMS sessions, which improves the usability of your IMS online system and the productivity of your users. It complements the existing IMS support for conversational transactions by managing the process of holding and resuming conversations for the end user. With IMS Multi-Dialog Manager, the active session can be placed on hold by pressing a single PF key and resumed later by pressing the same key. This new product uses standard IMS exits and does not require changes to IMS or application programs.

IMS Problem Investigator for z/OS, V1.1 (5655-K50) -

This new tool provides an enhanced level of problem determination services for IMS Transaction Manager and IMS Database Manager systems. These services include navigation aids, investigative procedures for IMS log data, and formatted and personalized reporting. Powerful automated features help to reduce the amount of time required to identify and analyze defects or other events of interest in the IMS log. Services can be accessed with an ISPF dialog or a batch reporting utility.

IBM IMS Performance Monitor for z/OS, V1.1 (5655-G50)

- This new tool provides real-time status monitoring and alerts for an active IMS subsystem. It detects active IMS subsystems when they are started and any IMS subsystems that might become active subsequently. Multiple IMS subsystems can be viewed from a single monitoring station, and these subsystems can be logically grouped however you choose, such as data sharing groups and shared queues groups.

IBM IMS Library Integrity Utilities for z/OS, V1.1 (5655-I42) -

This new tool includes all the features and functions of the IMS Library Management Utilities for OS/390®, which help you manage your IMS Access Control Blocks, PSB, and DBD libraries; and the IMS Advanced ACB Generator for OS/390, which contains additional features and functions to help you manage your IMS ACB libraries more effectively and provides a high-speed generation process. In addition, IMS LIU contains an Integrity Checker and Consistency Checker. The Integrity Checker protects your databases from corruption caused by using the wrong IMS control blocks for access. The Consistency Checker helps you ensure that the necessary definitions in your IMS subsystem have been created. The tool also provides easy-to-read reports and maps.

IMS Database Recovery Facility for z/OS, V2.1 (5655-I44) -

This is the follow-on version of IMS Online Recovery Service for z/OS (5655-E50) and contains all the features and functions of that product. It also adds several new key features.

IBM IMS High Performance Image Copy for z/OS, V3.1 (5655-K96) - This enhanced tool, formerly called IBM IMS Image Copy Extensions for z/OS, has been renamed. With the Image Copy function you can take an image copy for more than one database data set in parallel within a single job step.

IMS Connect for z/OS, V2.1 (5655-K52) - The new version features two-phase commit support for distributed environments across TCP/IP. A WebSphere Application Server in a non-OS/390, non-z/OS environment can now connect to IMS through IMS Connect (with IMS and IMS Connect initially in the same LPAR).

IMS High Availability Large Database (HALDB) Conversion and Maintenance Aid for z/OS, V2.1 (5655-K47) - This new version contains all the features and functions of IMS HALDB Conversion Aid for z/OS, V1.1. It extends the product by adding additional conversion options such as initial HALDB load assistance, status code 'BA' aid, and partition maintenance aid.

IMS High Performance Pointer Checker for z/OS, V2.1 (5655-K53) - This new version is designed to significantly improve performance and eliminate cumbersome job steps. It makes the product faster and easier to use.

IBM IMS Database Control Suite for z/OS, V3.1 (5655-L08) - This enhanced tool helps manage the day-to-day database environment. Version 3 provides significant performance improvements and faster navigation, and continues to evolve into a control center for IMS database management and maintenance.

IBM IMS High Performance Change Accumulation Utility for z/OS, V1.2 (5655-F59) - This enhanced tool allows you to run multiple change accumulation groups in parallel and streams output across all addresses simultaneously, completing jobs that used to take hours in a fraction of the time.

IBM IMS Extended Terminal Option Support for z/OS, V3.1 (5655-L51) - With IMS ETO Support V3.1, you can tailor all facets of ETO. It lets you set global options for the entire user community and override options for specific terminals or user IDs.

IBM Data Encryption for IMS and DB2 Databases, V1.1 (5799-F56) - This new tool is being offered as a PRPQ. It enables you to protect sensitive and private data at the segment level for IMS and at the table level for DB2. Encryption and decryption can be customized at these levels on the respective databases. The tool is implemented using standard IMS and DB2 exits. It exploits z/Series and S/390® Crypto Hardware features, resulting in low overhead encryption/decryption.

And finally, the new IMS Tool that is available through the Early Support Program (ESP) is:

IMS Online Reorganization Facility for z/OS, V1.1 - This tool provides new technology for reorganizing databases with minimal impact on database availability. The product operates as a Batch Message Program and copies the source database to the target database while application programs update the source database. As updates are made to the source database, update data elements are stored in a data space to be used after the copy process completes. After the target database is copied, the update data elements are used to locate the changed data in the source database and carry it forward to the target database. After nearly all of the updates (for example, 95%) have been applied:

The source database is quiesced
The remaining update data elements are processed
Database names are swapped
Index corrections are made
An Image Copy (IC2) is executed
The newly reorganized database is released for processing

As you can see, 2003 was filled with exciting news for IMS Tools. For 2004, we're going "full blast" with new tools and lots of updates for existing tools. Watch this space to get the very latest on what's new in 2004, and check our website as well.

A happy and prosperous 2004 to everyone!

For more information on these and all of IBM's DB2 and IMS tools, visit ibm.com/software/data/db2imstools. □

Margaret M Wilson

IMS Tools Market Manager
wilsonmm@us.ibm.com

Application Programming Recommendations and Considerations for OTMA Clients

OTMA provides an open interface to IMS/TM customers. With OTMA, a z/OS or TCP/IP application program can send a transaction or command to IMS without using SNA or VTAM. The number of programs that can currently connect to IMS/TM using OTMA is impressive: middleware software, gateway programs, database, even applications written by IMS customers. Each of the programs or applications that communicate with IMS using OTMA are considered OTMA clients. Figure 2 illustrates a number of OTMA clients.

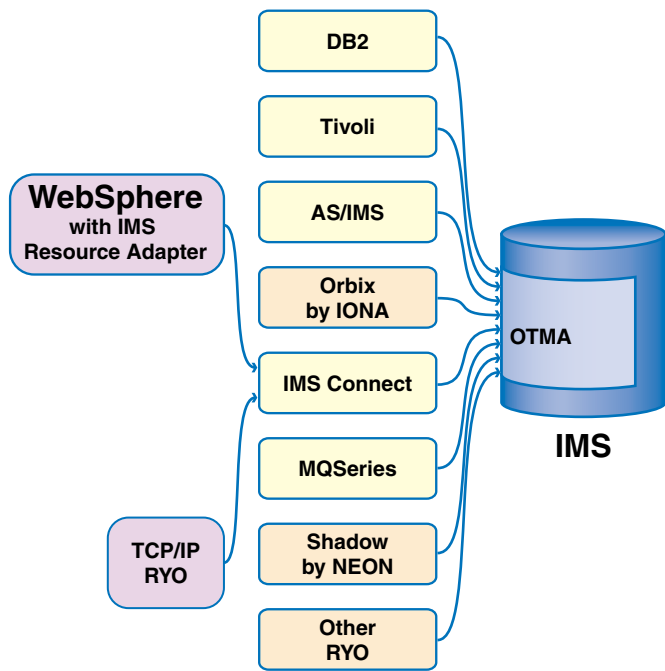


figure 2

The OTMA interface itself is very flexible. An OTMA client, an application program of the client, or both, can use OTMA in many different ways. The execution of some transactions can involve complex “handshaking” between IMS and the client program; some transactions can simply use the basic protocol. Figure 3 illustrates five ways that OTMA can be used to process an IMS transaction. This article provides recommendations and considerations for client applications so that an IMS transaction initiated by an OTMA client can be executed efficiently in IMS.

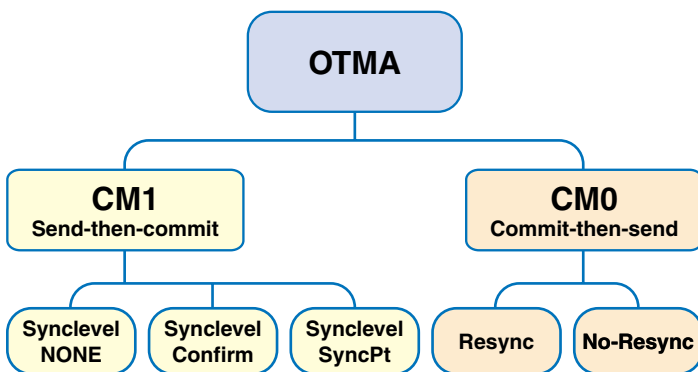


figure 3

Send-then-commit or Commit-then-send?

OTMA supports two different methods for controlling how IMS commits transactions: commit-then-send (CM0), which is the standard IMS commit flow, and send-then-commit (CM1). These are also known as asynchronous (CM0) and synchronous (CM1) processing.

For commit-then-send, IMS processes the transaction and commits the data before sending a response to the OTMA client. Input and output messages are recoverable.

For send-then-commit, IMS processes the transaction and sends the response to the OTMA client before committing the data. Input and output messages are non-recoverable.

An application program must decide which of these two methods to use. The following table can be used to help you decide which method is appropriate for your application.

Type of Processing	Commit-then-send (CM0)
Conversational transactions	Not supported
Fast Path transactions	Not supported
Remote MSC transactions	Supported
Shared queues	Supported in IMS V7 and above
Recoverable output	Supported
Synchronized Tpipes	Supported
Program-to-program switch	Supported

There are other factors that an application program might want to consider in deciding which method to use. For example, if the client is using MQSeries®, which is an asynchronous protocol, then commit-then-send, which is also asynchronous and supports synchronized tpipes, can be used. Alternatively, if database updates must be coordinated, then send-then-commit can be used. Careful consideration should be given to making the decision on which method to use.

Which CM1 Synclevel is Suitable to My Application?

If the application program uses send-then-commit, you must also decide which synchronization level, or “synclevel” to use. There are three choices:

- None - Output is sent, and no response from the client is requested. Data is committed if send is successful. Data is backed out if send fails.
- Confirm - Output is sent, and response from the client is requested. The OTMA client must respond with an ACK/NACK. Data is committed if ACK is received. Data is backed out if NACK is received.
- Syncpt - Output is sent, and response from the client is requested. The OTMA client must respond with an ACK/NACK. Data is committed if ACK is received and RRS commit is received. Data is backed out if NACK is received or RRS abort is received.

An application can decide, for example, that “inquiry” transactions should use synclevel=none, since there are no database updates, and that “update” transactions should use synclevel=confirm. In IMS Versions 6 and higher, the application can choose to use synclevel=syncpt to coordinate commit processing through RRS.

You Might Be Running with Too Much IMS Storage

If your application program can specify the client XCF member name, the OTMA tpipe name, or both, do not abuse IMS storage by having:

- different client XCF member names for every transaction invocation
- unlimited number of tpipe names created in your application

IMS tracks every client XCF member name and tpipe name and allocates storage for them, which remains in the IMS system until the system cold starts. It is highly recommended that your application program use only limited number of

client XCF member names and tpipe names to connect to IMS OTMA. For example, a tpipe name from an IMS Connect CMO message is a “client ID” name. If you write an application program modeled after IMS Connect that randomly generates a client ID for every input message, IMS could eventually have a storage shortage problem, because thousands of tpipes could have been created in the system.

In Summary

There are many decisions to make when designing an OTMA client. Hopefully this article provided some answers to help in making those decisions. Most of the material within this article can also be found in *Open Transaction Manager Access Guide and Reference*, and we hope to provide you with more answers in future IMS newsletters. OTMA provides many capabilities and benefits for an application program to easily access IMS data from any platform. With the proper application design, the possibilities are endless. □

Mike Artobello

IMS Developer

Jack Yuan

IMS Developer

2004 IMS Technical Conference

**September 27 - 30, 2004
Disney Hilton, Orlando, Florida**

In 2004, Orlando is the place to hear the latest and greatest about the IMS product family. Hear how IMS Version 9 is evolving with e-business on demand by providing product integration openness with tools for application development and connectivity. Get up to speed on the new facilities that IMS Version 9 offers to optimize IMS performance, capacity, availability, and recovery. Enhance your knowledge while enjoying the city that Walt Disney envisioned would be his greatest theme park creation...but evolved into one of America’s greatest convention cities.

When Walt Disney began purchasing land in the early 1950s, he envisioned an area in central Florida where he could establish a world-renowned entertainment area that could meet all family needs. What Mr. Disney didn’t envision was a world-class resort that also turned out to be one of the best destinations for conferences and conventions. The 2004 IMS Technical Conference will return to Florida once again and be a part of this exciting and entertaining city.

The Disney Hilton is located in the famous Walt Disney Resort Hotel area, and is adjacent to downtown Disney and minutes from the theme parks. Add to this the close proximity of Universal Studios and SeaWorld, and you have the unique opportunity to experience the nonstop entertainment Orlando has to offer. Many additional prominent attractions are an easy drive away, including Wet-N-Wild, Busch Gardens, and the Kennedy Space Center.

The IMS Technical Conference is an action-packed conference with non-stop learning during the day, but the evenings are available to enjoy all that Walt Disney envisioned over 50 years ago.

Bookmark our Technical Conference web site at www.ibm.com/services/learning/conf/us/ims for more details in the coming months. □

Amiet Goldman

Marketing Manager

The following new IBM Redbook™ for IMS is now available:

**Reorganizing Databases Using IMS Tools:
A Detailed Look at the IBM IMS High Performance Tools**
(SG24-6074)

Your database management system is the repository for your business data, so you need to make sure that data is consistent, well organized, and available. IBM IMS Tools has a collection of products for database reorganization that can help improve the availability of databases while they are frequently reorganized, improve the speed of numerous management tasks, and help in changing the database structure.

This IBM Redbook positions the different approaches for reorganizing IMS full-function databases and describes the usage of the IMS High Performance utilities set of products and IMS Parallel Reorganization product. The products covered by usage examples in this book are:

IBM IMS High Performance Unload Version 1 Release 1
IBM IMS High Performance Load Version 1 Release 1
IBM IMS High Performance Prefix Resolution Version 2 Release 1
IBM IMS Index Builder Version 2 Release 3
IBM IMS Parallel Reorganization Version 2 Release 1
IBM IMS Database Control Suite Version 3 Release 1

The other IMS related Redbooks previously published during the year 2003 are:

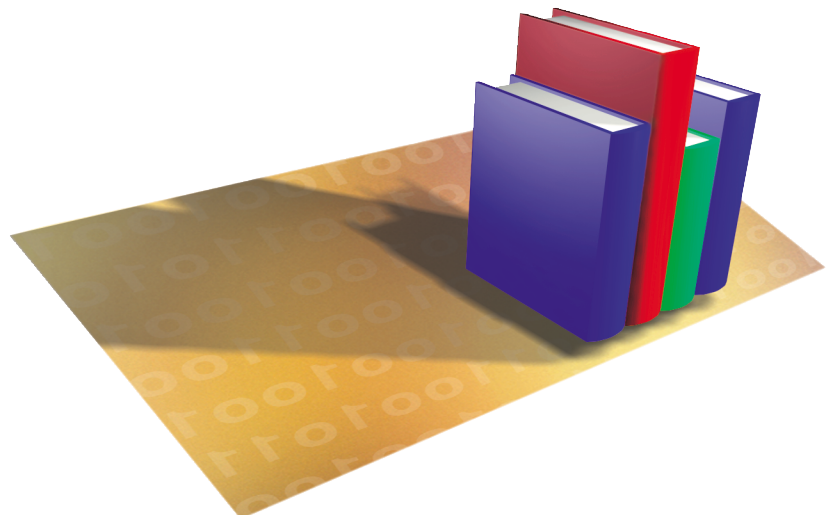
The Complete IMS HALDB Guide: All You Need to Know to Manage HALDBs
(SG24-6945), published 06/03.
Using IMS Data Management Tools for Fast Path Databases
(SG24-6866), published 01/03.

The famous IMS Parallel Sysplex Trilogy:

IMS in the Parallel Sysplex, Volume I: Reviewing the IMSplex Technology
(SG24-6908), published 03/03.
IMS in the Parallel Sysplex, Volume II: Planning the IMSplex
(SG24-6928), published 07/03.
IMS in the Parallel Sysplex, Volume III: IMSplex Implementation and Operations
(SG24-6929), published 07/03.

For more information on obtaining IBM Redbooks, joining a Redbook residency project, or downloading any Redbook, draft, or tip, visit the IBM Redbooks Web site at www.ibm.com/redbooks. □

Paolo Bruni
ITSO Data Management Project Leader



Hot Websites

IMS

www.ibm.com/software/data/ims

IBM Redbooks

www.redbooks.ibm.com

IBM Technical Conferences

www.ibm.com/training/conf/us/ims/

IMS Newsletter

www.ibm.com/software/data/ims/shelf/quarterly/

IMS Education

www.ibm.com/training

Reader Comment Form

www-3.ibm.com/software/data/rcf/

IMS Connector for Java

[www.ibm.com/software/data/db2imstools/imstools/
imsjavcon.html](http://www.ibm.com/software/data/db2imstools/imstools/imsjavcon.html)

IMS Support

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

IMS Tools

www.ibm.com/software/data/db2imstools/imstools

Editor-in-Chief: Glenn N. Galler
Creative Director: Rob Lee
Managing Editor: Sandy Sherrill
Senior Web Consultant: Michael Morrison
Customer Content Editor: Mandy Rooker

We welcome your feedback on any of the material contained in this issue.

To receive a free subscription to the **IMS Newsletter** please complete the form on the IMS Web page at www.ibm.com/software/data/ims/shelf/quarterly/.

or write to us in any of the following ways:

email: imsmtk@us.ibm.com

fax: (408) 463-4101

US Mail: IMS Newsletter

IBM Corporation

555 Bailey Avenue

DQY/A2

San Jose, CA 95141-1003



Do you want to receive the latest IMS announcements in your inbox? Go to the IMS homepage. Click on "Receive Recent IMS news" (under "Communities" on the right-hand side of the page). Fill out the form that pops up. Click on Submit Form. You will be added to an email distribution list that is growing every day!

They Must Be In Here Somewhere

It is a pleasure to read your feedback on the IMS Newsletter. We have a lot of interesting articles planned for this year and many of these topics come from your notes to me. I would like to encourage you to continue to provide me with ideas for future articles. If you are interested in telling your story on the cover of the IMS Newsletter, please let me know that too. Once again, the pictures of my two children are hidden in this issue of the IMS Newsletter.

We want to hear from you.

We will send an IBM IMS pin to every person who can find the two pictures of my children correctly in this issue.



You can download a copy of the IMS newsletter in PDF format from the IMS Web page at: www.ibm.com/software/data/ims/

If you would like to receive an IBM IMS pin, please e-mail your external postal address and the picture locations of my children to: ismkt@us.ibm.com

Or write to:

Editor-in-Chief, IMS newsletter
IBM Corporation
DQY/A2
San Jose, CA 95161-9023
U.S.A.

While you're at it, tell us what you think of the IMS Newsletter!

Glenn N. Galler
Editor-in-Chief



© International Business Machines Corporation
Printed in the U.S.A. 4/02
All Rights Reserved

DB2, DB2 Universal Database, IBM, IMS, MQSeries, MVS, OS/390, Parallel Sysplex, Redbooks, S/390, Tivoli, VTAM, WebSphere, z/OS and zSeries are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

The information in this document concerning non-IBM products was obtained from the suppliers of those products or from their published announcements. IBM has not tested those products and cannot confirm the accuracy of the 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.

GC26-9655-14



IMS Newsletter
IBM Corporation
555 Bailey Avenue
DQY/A2
San Jose, CA 95141-1003

PRESORT
FIRST CLASS
US POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1