

B91

IMS V9 QPP Experience migrating from
IMS V7 to IMS V9

Anders Öhrnberg

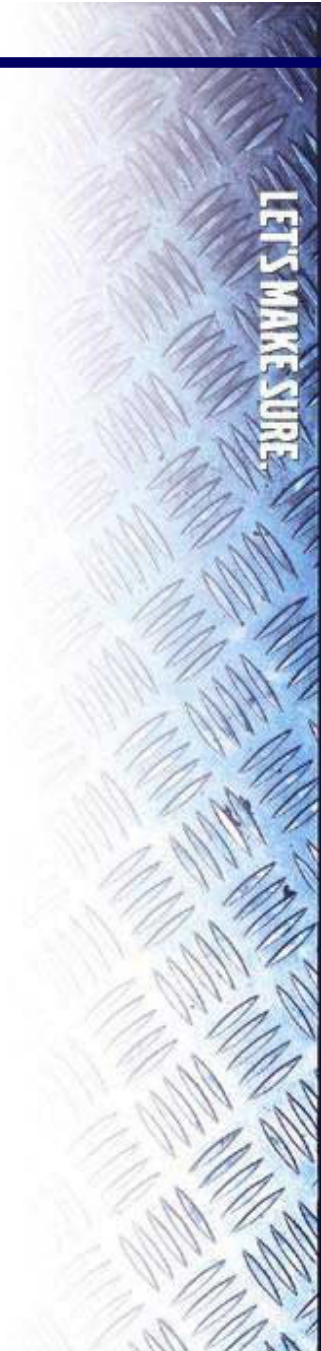
Volvo Information Technology

Gothenburg, Sweden

IMS
Technical Conference

Sept. 27-30, 2004

Orlando, FL



VOLVO



LET'S MAKE SURE.

NOTES PAGE

CUSTOMERS within AB VOLVO

Volvo Aero

Volvo Buses

Volvo Construction Equipment

Volvo Finance International

Volvo Information Technology

Volvo Parts

Volvo Trucks

Volvo Wheel Loaders AB

Volvo Penta

Celero Support

Volvo Logistics

CUSTOMERS not AB VOLVO

Volvo Cars

SCA

ATOS

LET'S MAKE SURE

- Overview Environment.
- Plans.
- Products and items tested.
- Implementation technique.
- CSL.
- Java.

LET'S MAKE SURE.

Volvo IT Z/OS sites:

Gothenburg (Sweden)

Greensboro (USA)

Allentown (USA)

Lyon(France)

Gent (Belgium)

LET'S MAKE SURE.

Overview IMS Environment use in IMS V9 QPP

VT01/IMZ1/IMV1

IBM CPU Model

2084-309

z/OS 01.04.00

IMS V9

IMx1CTL,DRC,DLI

IMx1M*

IMx1SCI, OM

IMx1CQS

IMx1IMF

IMx1ICV1

VT03/IMZ3

IBM CPU Model

2084-307

z/OS 01.05.00

IMS V9

IMZ3CTL,DRC,DLI

IMZ3M*

IMZ3SCI, OM

IMZ3CQS

IMZ3IMF

IMZ3ICV1

LET'S MAKE SURE

NOTES PAGE

Two members in a shared queues group. One with V9 and one with V7. Jobs (DLI batch, BMPs) executing in any lpar, always on the IMS level that exists in that lpar. One lpar with Z/OS 1.5 and one with Z/OS 1.4.

LET'S MAKE SURE

IMPLEMENTATION PLANS

- Initial install and implementation tech system 20 Jan 2004
- IMZ3 test tech system 20 Feb 2004
- IMZ1 test tech system June 8 2004.
- IMVT customer test system June 8 2004.
- Application development system (2-3 Week's) September 2004.
- Production April 2005.

LET'S MAKE SURE.

NOTES PAGE

Products ready for IMS V9 at the time IMS V9 were implemented in IMZ3:

Deltaplex	1.3.04
ETA	3.0.03
ACF2/IMS	6.5

We needed ofcourse V7 DBRC Migration/Coexistence APAR/PTF, PQ72838/UQ82308 on the system. This made us put on 40 other ptf's wich affected many modules in RESLIB. So prepare in advance. Most of them are for SQ and HALDB.

LET'S MAKE SURE

Products IMPLEMENTED and TESTED May 2004

Deltaplex	1.3.04	
ETA	3.0.03	
ACF2/IMS	6.5	
Mainview for IMS Online	4.1.00	
Autooperator for IMS	6.3.0	
BTS	3.1	
HPC/HPU	2.1	Pointer checker/Unload
MAXM Reorg/EP	1.5.00	
IMS Connect	2.1	

No major problems has occurred during the testing of listed products so far.

LET'S MAKE SURE.

Test items for Volvo IT.

V8 CSL, plex commands etc

IVP

HALDB Online Reorganization

Syntax Checker

IMS JAVA DB2 Interoperability

XML DB Decomposed and Intact storage

IMSPLEX DB Commands

IMS Control center

LET'S MAKE SURE.

NOTES PAGE

The CSL feature seems to a very nice tool. Gives you a chance to do for example a web interface for commands using classic and plex commands.

LET'S MAKE SURE

What happened so far ?

IMS Library management utility replaced with freeware from NEON systems. (DBD MAP).

Mainview conflict between releases.

SPOC not user friendly but rexx interface can help you.

XML DB

JAVA DB2

LET'S MAKE SURE

NOTES PAGE

LMU has to be replaced with NEON freeware.

Problem generating schemas with DLI model utility when fields were not in DBD.

JAVA region JMP could not connect to DB2. JAVA-regions with DB2-connectionabend 0C4 at restart if dd-card DFSDB2AF is added in the procedure.

LET'S MAKE SURE

IMPLEMENTATION TECHNIQUE

We upgraded recon when IMS was up and running.

We are using an N+n technique for most of the IMS and vendor product datasets used with IMS.

Z/OS Symbolic variables used with Symbolic aliases.

It executes towards the **right maintenance level** by using the aliases created with the Symbolic Alias Facility. For example F1IMZ0.IMS.USERLIB point to physical dataset

for example:

F1IMZ0.IMS.V701P005.USERLIB in LPAR VT01 or

F1IMZ0.IMS.V901P001.USERLIB in LPAR VT03.

When upgrading IMS to a new release of IMS, any user written exits that use IMS macros should be re-assembled with the macros from the new release. This means that separate load libraries for these exits and other upgraded modules will need to be maintained for each release or upgrade.

IBM has provided a way to upgrade one Lpar at the time (Which means one IMS in a Data Sharing group (Shared queues group) at the time. Jobs and STCs can still run in any Lpar but runs on the version of the product that exists in the Lpar it is executing.

You can use one IEASYMxx member for the whole Sysplex and one member per Lpar. The Lpar members are the ones that controls the whole maintenance flow. If a variable has a value at IPL, then that will be the value used in that Lpar. So implementation plans and preparations have to be carefully done.

Symbolic Alias Facility provides the ability to have an alias, but have that alias resolve to different data set names depending on which system the data set is allocated on. It does this by allowing you to place a system symbol in the alias definition.

```
DEFINE ALIAS (NAME(SYS1.F1IM00.IMS.RESLIB))-  
SYMBOLICRELATE(SYS1.F1IM00.IMS.&IMSVERS..RESLIB (*
```

Initially, the symbol &IMSVERS would be set to V701P005 on all systems. When you

are ready to test the new release on one of the systems, you would change the &IMSVERS symbol *on that system only* to V901P001. Thus, you can have a shared catalog, and have users on two systems using the same data set name, but have each user executing on a different data set.

IMPLEMENTATION TECHNIQUE

At upgrade changes are made to ACBLIB, RESLIB, MODBLKS, MACLIB, USERLIB (Exits, modifications etc).

SYS1.PARMLIB members APF list, MLPA list, LINK list.

LET'S MAKE SURE

NOTES PAGE

MLPA

```
INCLUDE LIBRARY(SYS1.F1IM00.IMS.&IMSVERS..RESLIB)
      MODULES(DFSMRCL0,DFSAFMD0)
```

APF List

```
APF ADD DSNAME(SYS1.F1IM00.IMS.&IMSVERS..RESLIB)          SMS
```

LINK List

```
LNKLST ADD NAME(VT1.PUO00B) DSNAME(SYS1.F1IM00.IMS.&IMSVERS..RESLIB)
```

LET'S MAKE SURE

IMPLEMENTATION TECHNIQUE

This is the DD card in both the IMZ1 and IMZ3.

```
DD DISP=SHR,DSN=F1IM00.ACF2.LOAD  
DD DISP=SHR,DSN=F1IM00.DLP.LOAD  
DD DISP=SHR,DSN=F1IM00.ETA.LOAD
```

This is what it looks like in IMZ3CTL in VT03.

```
IGD104I F1IM00.ACF2.V901P001.LOAD  
IGD104I F1IM00.DLP.V901P001.LOAD  
IGD104I F1IM00.ETA.V901P001.LOAD
```

This is what it looked like in IMZ1CTL in VT01 with V7.

```
IGD104I F1IM00.ACF2.V701P005.LOAD  
IGD104I F1IM00.DLP.V701P005.LOAD  
IGD104I F1IM00.ETA.V701P005.LOAD
```

NOTES PAGE

Here is a sample from STEPLIB from both lpar and both IMS systems. The DD cards point to the same ALIAS. It is an alias and no physical dataset.

What you see being allocated in each job is the physical dataset, where the symbolic variable has been resolved to the value set for that specific lpar.

In this case IMS V7 in VT01 and IMS V9 in VT03.

Common Service Layer

Three new STCs. (New messages to handle.)

SCI **Structured Call Interface.**

OM **Operation Manager.**

RM **Resource Manager.**

Features we are looking at exploring:

IMS Control Center (using IMS Connect).

TSO SPOC plexcommands ?

OM API (Using Rexx)

WEB/JAVA Command interface

LET'S MAKE SURE

NOTES PAGE

Standardizes communication between IMSplex members cross-memory services on same LPAR, XCF between LPARs.

We are using SA (System Automation) and need to be able to handle messages in a proper way to be able to tell SA when the STC is up or down.

STARTUP MESSAGES CSL0020I OM READY IMZ3OMOM
 CSL0020I SCI READY IMZ3SCSC

SHUTDOWN MESSAGES CSL0021I OM SHUTDOWN COMPLETE IMZ3OMOM
 CSL0021I SCI SHUTDOWN COMPLETE IMZ3SCSC

LET'S MAKE SURE

Common Service Layer

IMZ0 IMS Single Point of Control

Command ===>

Plex . Route .

Enter '/' to view command response, 'i' to reissue a command, o a command.

Act	Status	Command
	Complete	QUERY TRAN NAME(VP12*) SHOW(CLASS,STATUS)
	Complete	QUERY TRAN NAME(V*) SHOW(CLASS,STATUS)
	Complete	QUERY TRAN NAME(V*) QCNT(GT,0)

***** Bottom of data *****

LET'S MAKE SURE

NOTES PAGE

Plex commands will take time for programmers to learn. It would be nice to be able to set up commands for a specific customers programmer in advance.

(Also see security later on in this presentation).

If you could set up a panel with commands useful for the programmer, it would save so much time in learning the new command syntax.

LET'S MAKE SURE

Common Service Layer

IMZ0 IMS Single Point of Control

Command ===>

Plex . Route .

Enter '/' to view command response, 'i' to reissue a command, o a command.

Act	Status	Command
	Complete	QUERY TRAN NAME(I*)
	Complete	QUERY TRAN NAME(I*)
	Complete	QUERY TRAN NAME(I*)
	Complete	QUERY TRAN NAME(I*)
	Complete	QUERY TRAN NAME(VCMD) SHOW(CLASS,STATUS)
	Complete	UPD TRAN NAME(VCMD) SCOPE(ALL) SET(CLASS(4))
	Complete	QUERY TRAN NAME(V*) SHOW(CLASS,STATUS)
	Complete	QUERY TRAN NAME(V*) QCNT(GT,0)

***** Bottom of data *****

LET'S MAKE SURE

NOTES PAGE

Here you can see how commands are stacked even if they are exactly the same command you just issued.

This is off course to be able to show the result from each command

LET'S MAKE SURE

Common Service Layer

CSL

IMS **IMZ3CTL**

DFSPBX**Z3** CSLG=**XZ3**

DFSCGX**Z3**



IMSPLEX=**IMZ0**

CQS **IMZ3CQS**

CQSIPX**Z3**

IMSPLEX(NAME=**IMZ0**)

IMSCONNECT **IMZ3ICV1**

HW**Z3ICV1** (HWSCFGxx)

IMSPLEX MEMBER=**IMZ0**

TMEMBER=**IMZ3SCI**

SCI (Structured Call Interface) IMZ3SCI
CSLSIX**Z3** IMSPLEX(NAME=**IMZ0**)

RM (Resource Manager) IMZ3RM
CSLRIX**Z3** IMSPLEX(NAME=**IMZ0**)

OM (Operations Manager) IMZ3OM
CSLOIX**Z3** IMSPLEX(NAME=**IMZ0**)

LET'S MAKE SURE

NOTES PAGE

You can see how our namestandard makes it is easy to follow the datasharing group.

It was not that easy to find all the parameters for CSL. They where not all in the CSL manual as you could have hoped.

To be able to use IMS control center through IMS connect we needed to specify parameters for CSL in the IMS Connect parameter member HWSCFGxxx.

LET'S MAKE SURE

Common Service Layer

Example of IMS PROCLIB DD

DFSINTZ0

DFSPBXZ1

DFSPBXZ3

DFSSPMZ0

DFSSQXZ1

DFSSQXZ3

DFSVSMZ0

F1IMZ0.IMS.V701P005.PARMLIB

IMS V7

F1IMZ0.IMS.V901P001.PARMLIB

IMS V9

NOTES PAGE

We use a shared DD PROCLIB dataset between all members in the datasharing group.

The name standard used makes it easy to recognize what system the members belongs to.

We also use the N+n standard so we can easaily upgrade the PROCLIB for the next realese/version.

Common Service Layer

Security för commands in ACF2 för OM (SPOC)

A lot to specify for each command.

Missing resources level in authority for commands.
(For example: Transactions starting with R*).

IMS.IMZ0.STA.TRAN should be IMS.IMZ0.STA.TRAN.R-

LET'S MAKE SURE.

NOTES PAGE

If you want to protect all command there will be a lots of typeing.

Example: The STO command can be protected using wildcards `IMS.IMZ0.STO.*` but you probably would like to protect STO DC, STO WADS, STO SUBSYS etc.

To be able to set up an environment for a specific company/department and the resources they are using the RACF/ACF2 interface should support resource level. If a company has all programs, transactions and DBs starting with R*, it would be nice to allow them to make STA, STO etc command ONLY to resources starting with R-. Example from the resource OCM in RACF/ACF2 class OPERCMDS. `IMS.IMZ0.STA.TRAN` should be `IMS.IMZ0.STA.TRAN.R-`

LET'S MAKE SURE

JAVA and IMS

IMS Java tested with JVM 1.3 and will also be tested with JVM 1.4.

JDBC to DLI.

Java to XML DB.

Java to DB2.

Reading from the IOPCB.

IMS Java Remote Data services ?????

LET'S MAKE SURE.

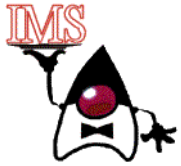
NOTES PAGE

When a JAVA programmer starts programming JAVA programs that will execute in IMS, they will not know anything about IMS.

The environment they will use should also be that way. Reading from the IOPCB is one of the things **Odd Olsen** at Volvo IT is solving for the future JAVA programmers.

With the new JAVA processors and the new charging for these processors JAVA becomes more interesting.

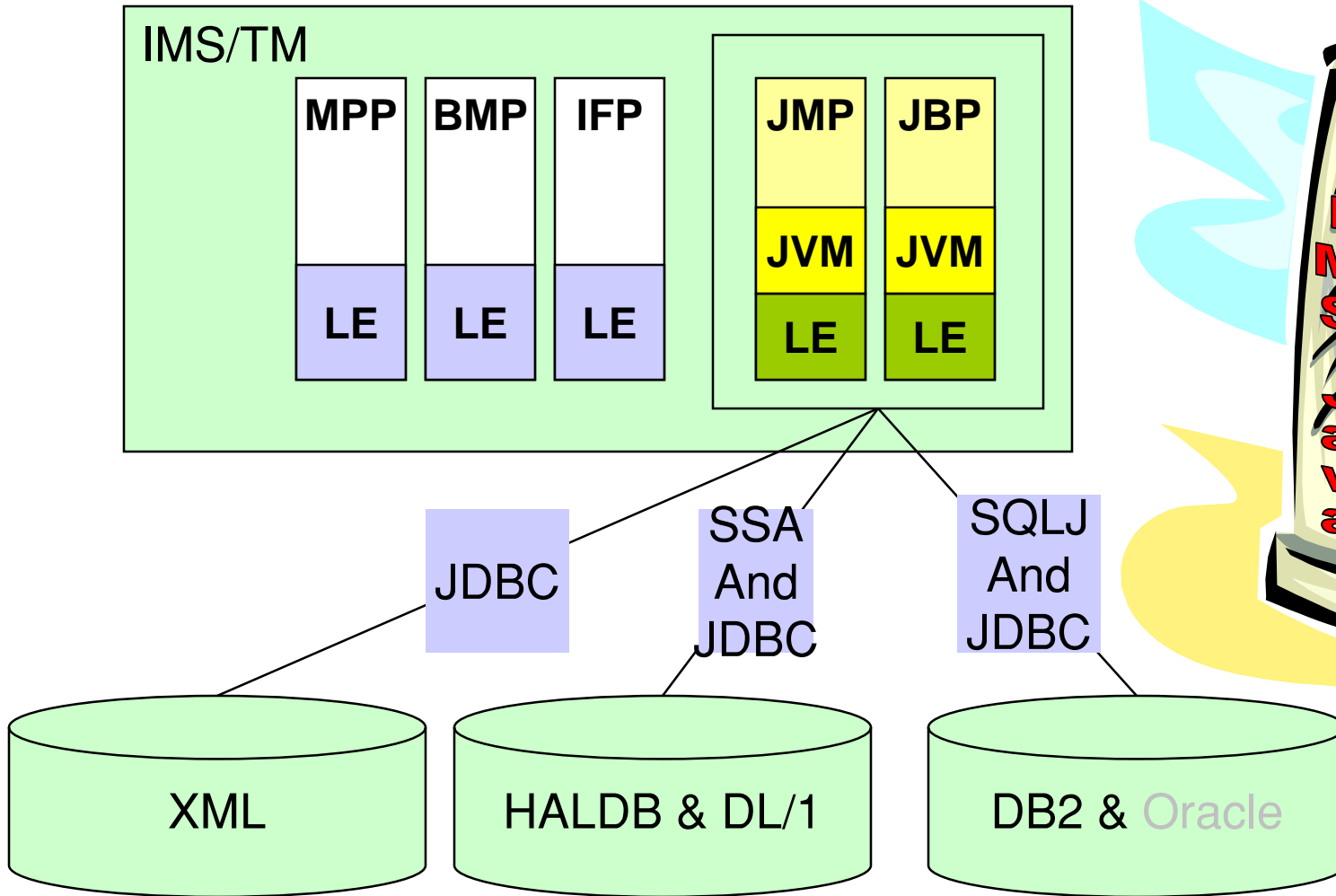
LET'S MAKE SURE.



Next Generation IMS Systems

IMS Java Transaction Programs

LET'S MAKE SURE.

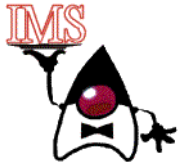


Benefits with IMS Java

- The new Java regions supports all databases in z/OS - XML/DB, HALDB, DL/1 and DB2
- Support for both JDK 1.3.1 and JDK 1.4.1
- The new Java regions can also run the new object oriented Cobol

Why start with IMS Java

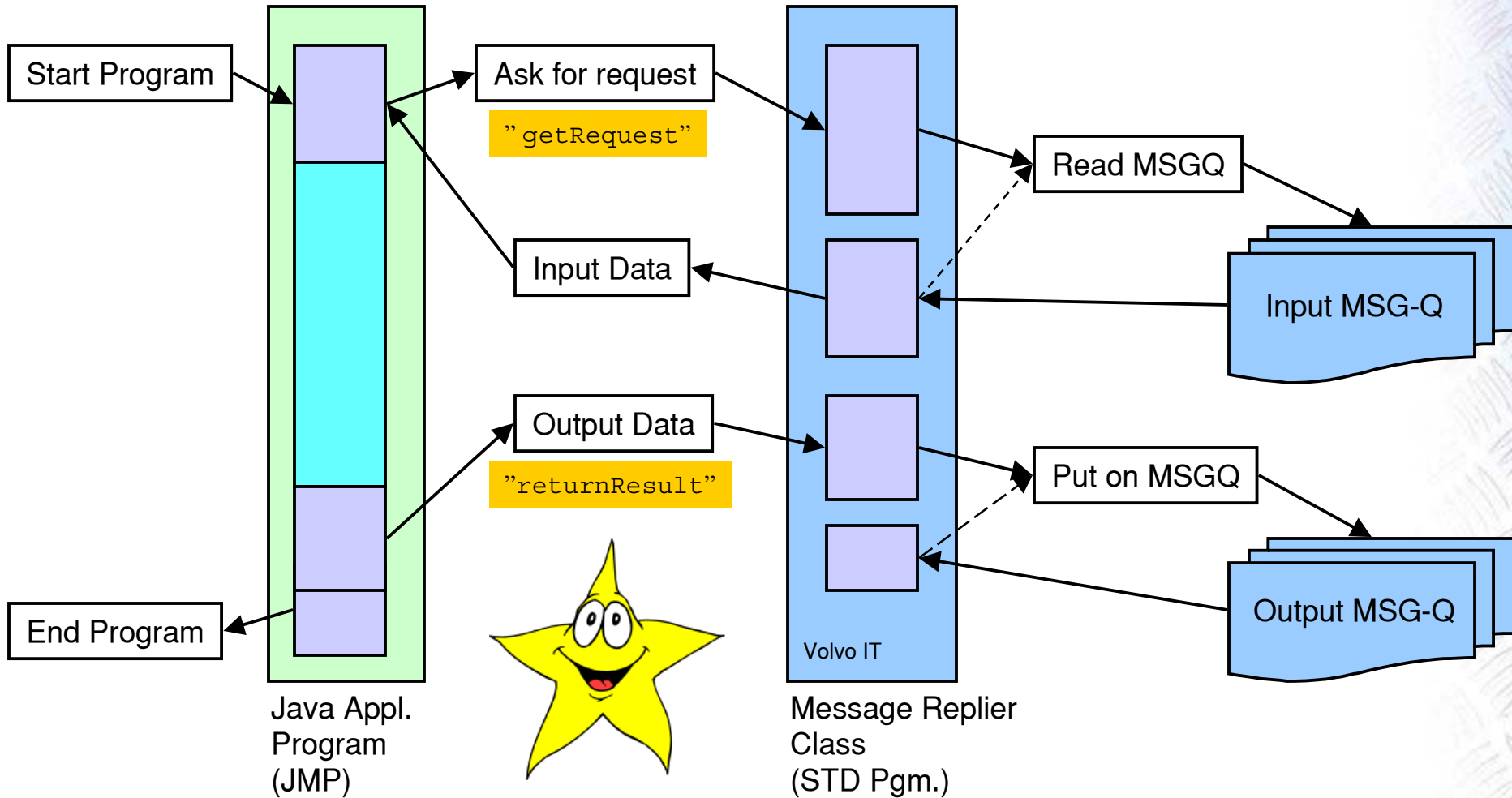
- Colleges teach Java, very few still teach COBOL
- Colleges teach relational DBs with SQL access, very few teach hierarchical access with SSA access
- You get access to a real XML-database.
- You can use both JDBC and SSA to access DL/1, or HALDB data, depending on your skills
- Customers requests application servers with Java support
- Rapid Application Development with object oriented programming and new programming tools
- If you know IMS, it's not a huge step to start programming IMS Java
- If you use IMS today, the changes in the infrastructure is reasonable



Next Generation IMS Systems

Using ImsMessageReplier with IMS Java

LET'S MAKE SURE.



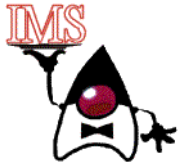
NOTES PAGE

IMS Message Replier

IMS Message Replier is a standard program that's makes it easy to write IMS transaction programs.

- The message replier provides a general interface to exchange messages with IMS. Today we only support IMS but in the future we can also support VCOM, MQ etc.
- Makes it easier to write. Takes care of all the mumbo jumbo when writing IMS Applications
- Today you can send/receive data in a Byte array or String
- You can easily keep the business code independent of the application server

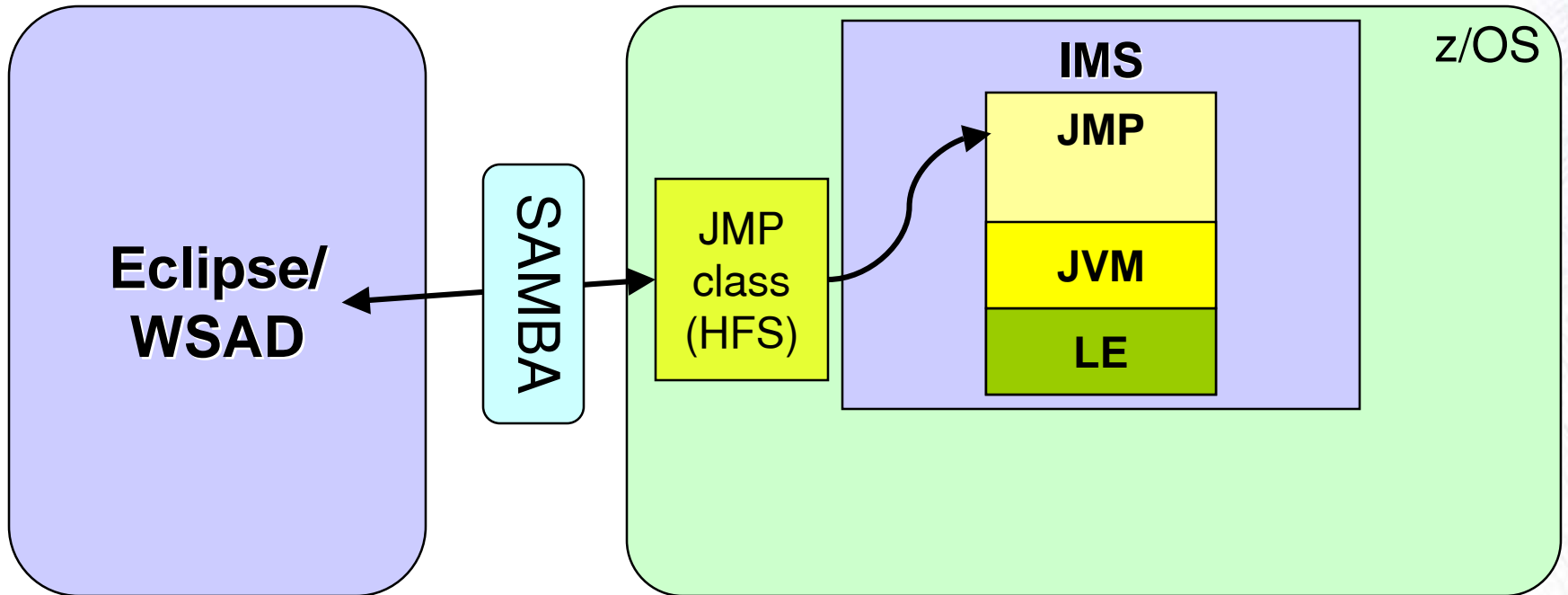
LET'S MAKE SURE



Next Generation IMS Systems

Developing IMS Java with ECLIPSE/WSAD

JDK 1.3 or JDK1.4



Third party packages

- VDIS
- Apache
- IBM
- IMS Java
- etc.

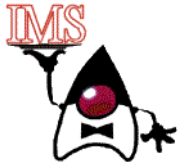
LET'S MAKE SURE

NOTES PAGE

System Development

- Use Eclipse as your development tool
- Edit the files directly in HFS on z/OS using Samba
- The files are automatically “compiled” to classes by Eclipse
- IMS have access to the classes and can run them directly
- Supporting both JDK 1.3 and 1.4
- All development is done in the target environment
- You can use almost any third party packages
- Deployment is just a move of classes from then test IMS to the prod IMS system (just as with Cobol programs today)

LET'S MAKE SURE



Next Generation IMS Systems

IMS Java Program Code

```
import com.volvo.ims.jump.ImsExit;
import com.volvo.ims.jump.ImsMessageReplier;

public class TransTemplate {
    public static void main(String args[]) {
        try {
            ImsMessageReplier q = new ImsMessageReplier();
            String mid = q.getRequestString();
            String mod = mid.toUpperCase().trim();
            q.setResponse(mod);
        }
        catch (Exception e) {
            ImsExit.abend(e, "The program has abended");
        }
    }
}
```

LET'S MAKE SURE

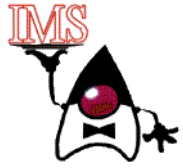
NOTES PAGE

IMS Java Program

This is an example of a IMS Java program. Just a plain java program using the `ImsMessageReplier` class to read and write messages from/to the IMS message queue and the program runs under an IMS transaction.

Writing a transactional program in IMS Java is as easy as writing a standard Java program.

LET'S MAKE SURE.

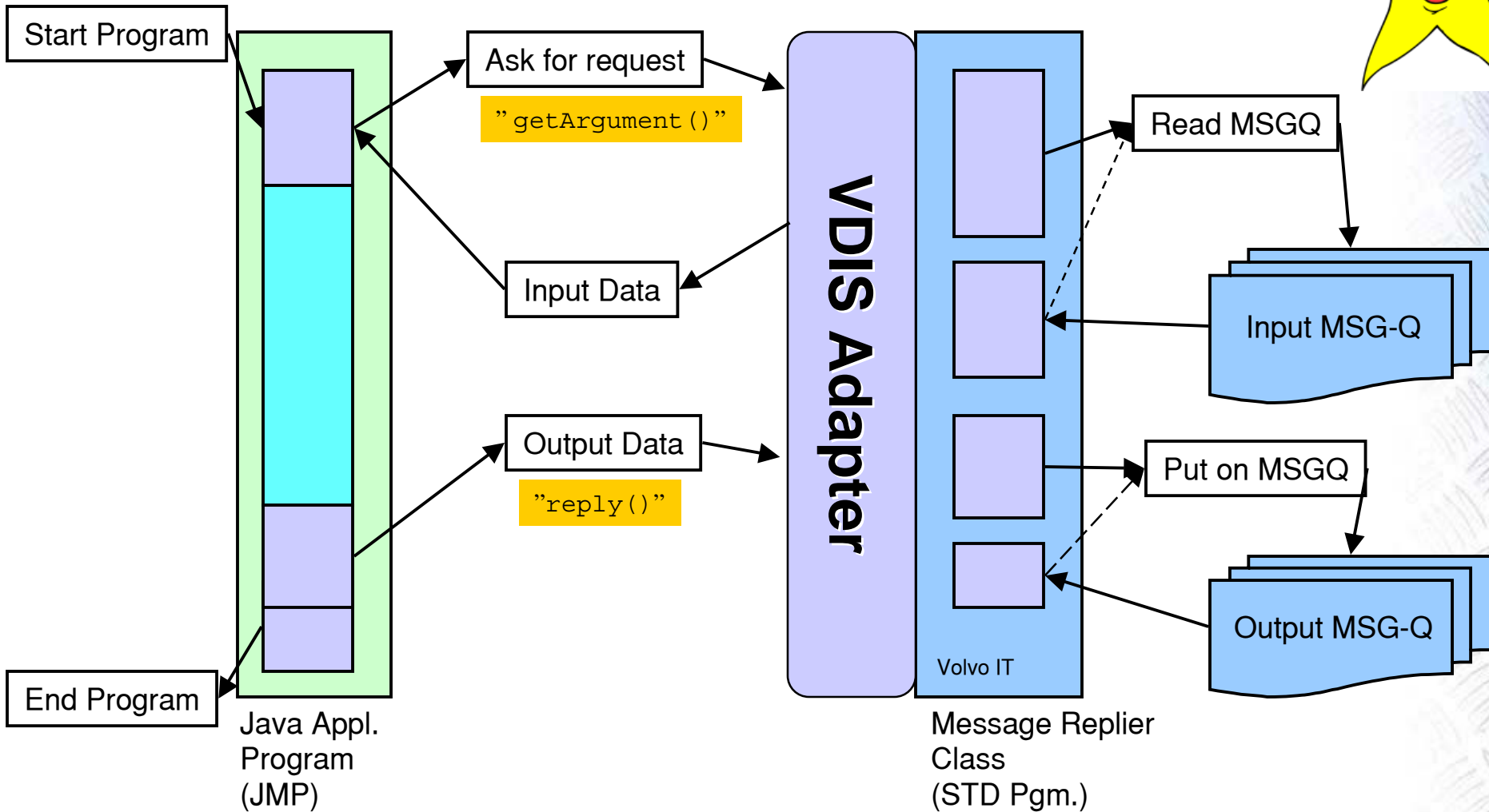


Next Generation IMS Systems

Using VDIS with IMS Java



LET'S MAKE SURE



NOTES PAGE

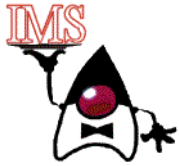
VDIS Adapter

If you feel like using VDIS, There's a VDIS adapter that makes it possible to Use the VDIS Dispatcher in IMS Java

VDIS makes it possible to send messages in XML-format that is transformed into Java value objects that can be used by the IMS Java program.

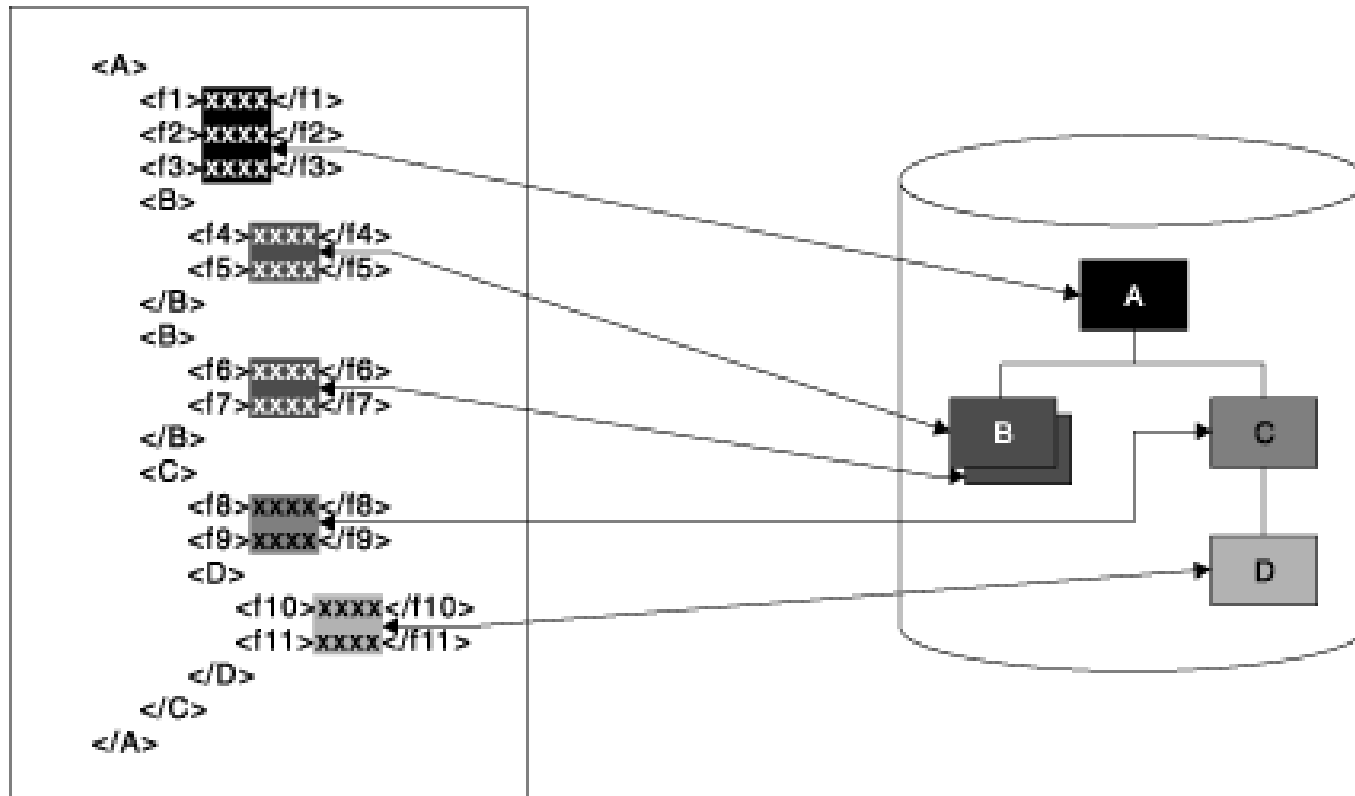
NOTE! A corresponding solution is available in Cobol

LET'S MAKE SURE



Next Generation IMS Systems

XML Database



LET'S MAKE SURE.

NOTES PAGE

XML Databases

Because XML and IMS databases are both hierarchical, IMS is a natural DBMS for managing XML documents. IMS allows you to easily receive and store incoming XML documents as well as compose XML documents from existing, legacy information stored in IMS databases. For example, you can:

- Compose XML documents from all types of existing IMS databases, to support, for example, business-to-business on demand transactions and intra-organizational sharing of data.
- Receive incoming XML documents and store them in IMS databases. These databases can be legacy databases, or new databases. XML documents are stored decomposed, where the document is parsed and element data and attributes are stored in fields in segments as normal IMS data. This is appropriate for what are sometimes called data-centric documents.

IMS Java has two SQL99 extensions for User Defined Functions (UDFs): retrieveXML and storeXML. These UDFs are used during JDBC calls to store and retrieve XML from IMS databases. This interface is independent of the physical storage of the data.

LET'S MAKE SURE

Anders Öhrnberg

Volvo IT

IMS Local Technical Support

Dept 8295, DA1S

S-405 08 Gothenburg

Tel: +46 31 322 43 02

E-mail: **Anders.Ohrnberg**@Volvo.com

LET'S MAKE SURE