



Virtual IMS user group: Newsletter 88



Welcome to the Virtual IMS user group newsletter. The Virtual IMS user group at itech-ed.com/virtualims is an independently-operated vendor-neutral site run by and for the IMS user community.



Figure 1: Open Database Components

Virtual IMS user group presentation

The latest webinar from the Virtual IMS user group was entitled, "Distributed access to IMS via Java". It was presented by Joseph Leamon, IMS Software Engineer (UDB) at IBM and Mahmood Abuzaina, Software Engineer at IBM.

Joseph is an IMS Software Engineer, focusing on providing support and

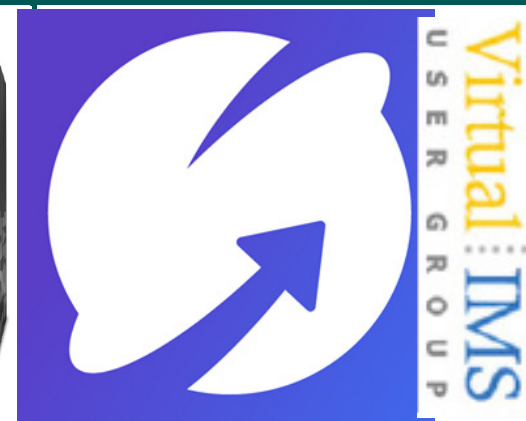
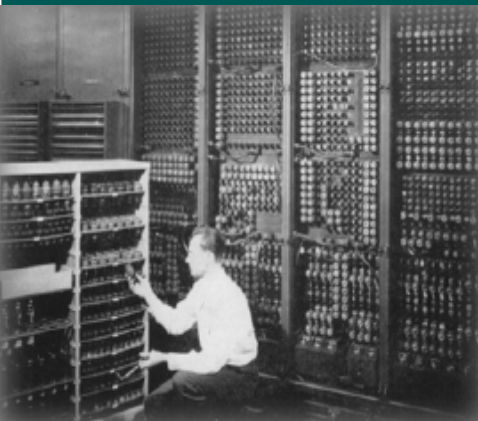
new functions for the IMS Universal Drivers. Joseph graduated from San Jose State University and has been working at IBM for over 7 years.

Mahmood is a software engineer at IBM. He has been working with IMS for around 5 years where his main focus has been on the IMS Java Universal Drivers. He graduated from San Jose State University with a BA in Software Engineering.

Joseph Mahmood started his presentation by giving an overview of IMS Open Database. He said that

Contents:

Virtual IMS user group presentation	1
Meeting dates	6
Sponsorship opportunity	6
IMS articles and blogs	6
About the Virtual IMS user group	6



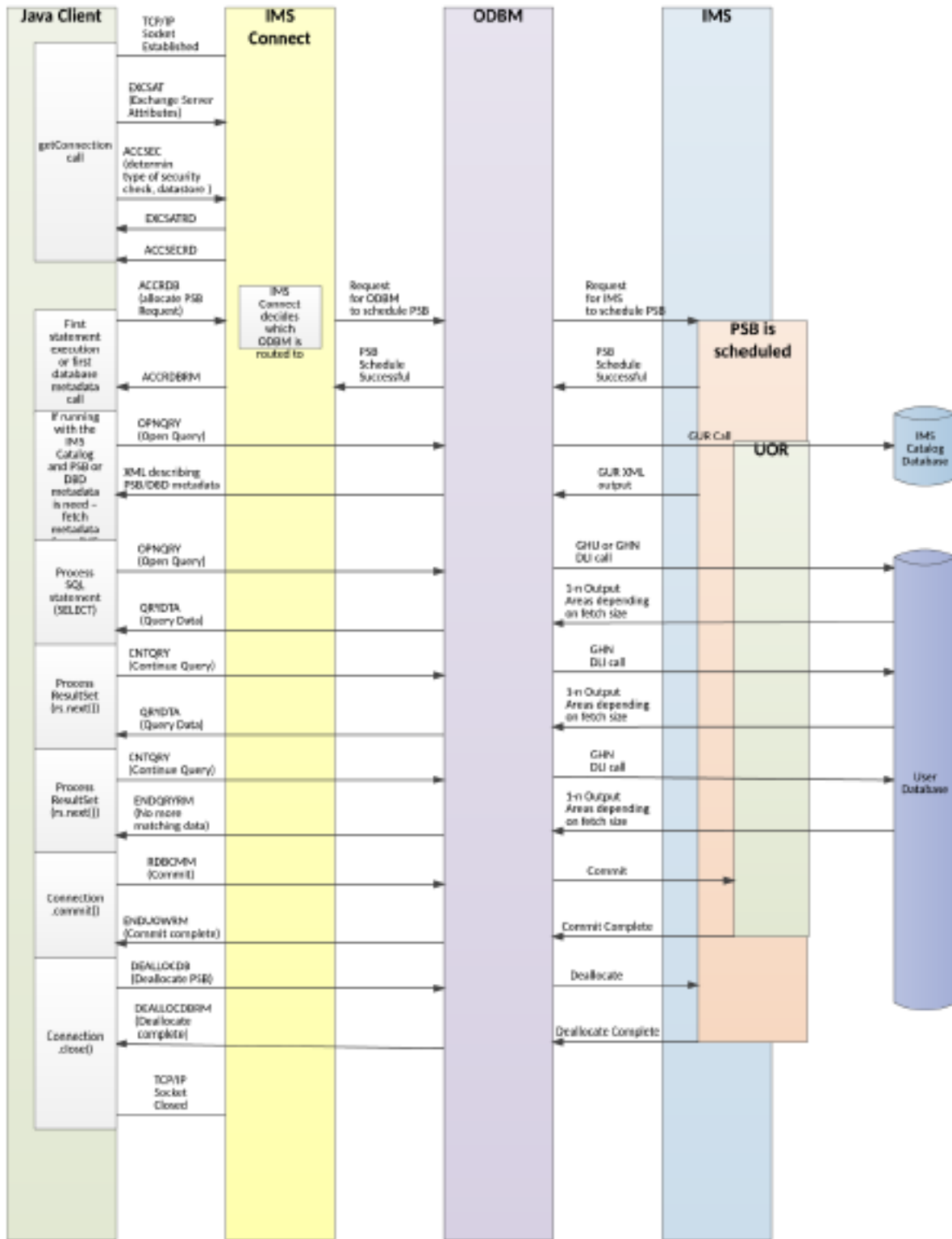
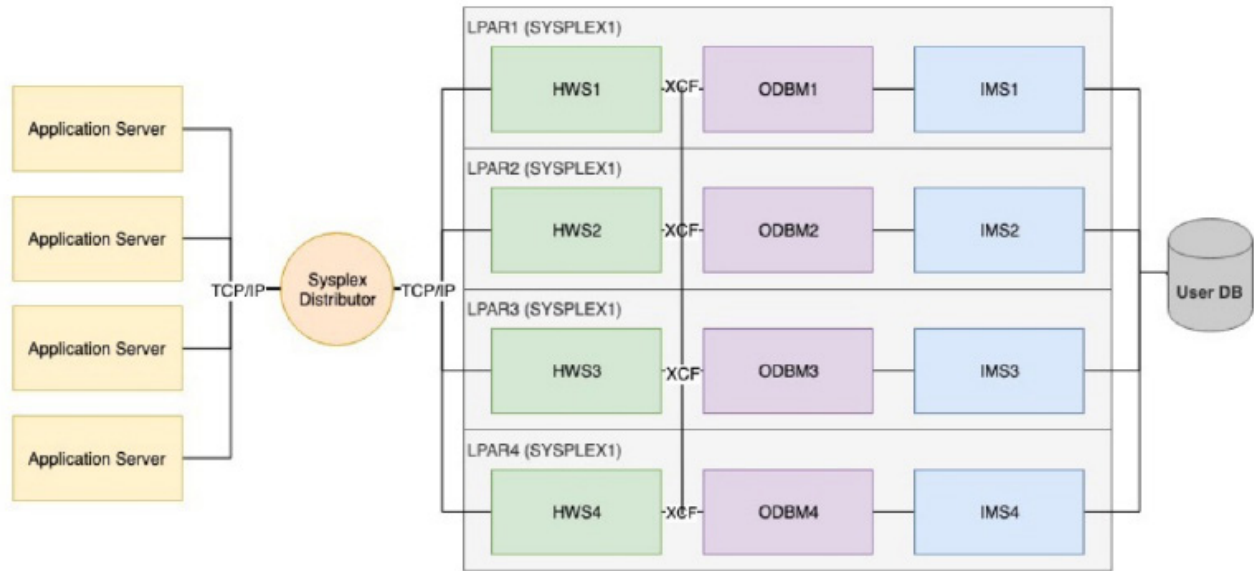


Figure 2: Understanding Open DB communication flows

Open Database provides an environment that manages access to online IMS

databases from anywhere in the enterprise. Customers

are leveraging it to solve business requirements



45

Figure 3: Open DB High Availability

by allowing: distributed application developers to access IMS data directly; analytics tooling and platforms to access IMS data directly; data analysts to inspect, resolve, and maintain data; DBAs, system programmers, and application developers to gain insight into IMS schemas, metadata, and data.

Figure 1 shows the Open Database components. IMS Universal Drivers provide Java-based APIs to manage connectivity and access to IMS databases. Two APIs are provided: JDBC Interface (SQL) and Java DL/I Interface. IMS Connect is the IMS endpoint for accessing IMS data over

TCP/IP. It works with ODBM as a DRDA (Distributed Relational Database Architecture) server. DRDA is a communications protocol for database servers. Open Database Manager (ODBM) works with IMS Connect as an IMS DRDA server. ODBM is responsible for issuing DL/I requests on behalf of the distributed application and returning the results.

In terms of infrastructure, IMS must be part of an IMSplex. The Structure Call Interface (SCI) is a Common Service Layer (CSL) address space for establishing an IMSplex for intercommunication between IMS components. Operations Manager (OM)

is a CSL address space for processing commands and routing them to various IMS components (Type-2 commands and some Type-1 commands). Open Database Manager (ODBM) is a CSL address space for processing distributed requests that access IMS data. IMS Connect (ICON) is the TCP/IP gateway for IMS. It works with ODBM as a DRDA server for distributed requests that access IMS data. IMS Universal Drivers (UDB) are client-side Java APIs to manage connectivity and request data from IMS databases.

Mahmood then looked in detail at the JCL needed to use these components.

IMS ships a Java library, `imsudb.jar`, which is used by the Open DB client application to access IMS data using the JDBC interface which interprets SQL into DL/I requests, and to access IMS data using the DL/I for Java interfaces. IMS ships four IMS Universal Driver Resource Adapters, which are:

- `imsudbJLocal.rar` – JDBC interface with single-phase commit (recommended)
- `imsudbJXA.rar` – JDBC interface with two-phase commit (recommended)
- `imsudbLocal.rar` – CCI interface with single-phase commit
- `imsudbXA.rar` – CCI interface with two-phase commit.

All parts are shipped as part of the IMS Java FMID (`JMKxx06` where `xx` is the IMS release level). There's no additional charge on top of the IMS base.

Figure 2 shows the Open DB communication flows. Mahmood went on to explain exactly what happens at each stage. He said that this information can be used to

understand the Open DB client application requests, and understand the time taken to make requests.

Mahmood Abuzaina next explained why it is important to take into account high availability when planning to deploy Open DB infrastructure. He explained that IMS Connect can communicate with one or more ODBMs, either on the same LPAR or different LPARs within the same SYSPLEX. Also, an ODBM can communicate with one or more IMSs within the same LPAR. That means multiple ODBMs can communicate with a single IMS. He said that customers who have implemented High Availability Open DB environments use SYSPLEX Distributor to load balance TCP/IP communications from Open DB client applications to IMS Connect. IMS Connect routing exists to determine routes between IMS Connect and ODBM. Customers have at least one ODBM per IMS Control Region. The set-up is illustrated in Figure 3.

Mahmood explained that in order for Open DB client applications to use PSBs to access IMS databases, the PCBs must have a PCBNAME or a PCB label. The Open DB solution uses AIBTDLI or AERTDLI interfaces, which require that the PCBs are referenced

by PCBNAME or PCB label rather than address/offset. It's important to ensure that PCBs have an appropriate level of processing option (PROCOPT) for the distributed application because this can: reduce locking; limit access to the database; and limit database updates when necessary. It's also important to ensure that the PSB is secured under the appropriate RACF resource class to limit access, audit, account for access to the databases.

Mahmood then gave some examples of Java JDBC and DL/I applications.

Next, he said that the IMS Universal Database drivers rely on metadata, ideally from the IMS Catalog. Metadata is used to: understand segment relationships (based on DBD definition/hierarchy); understand column/field layouts within a segment (FIELD/DFSMARSH macro statements in the DBD source); and to understand how to interpret column/field data types (FIELD/DFSMARSH macro statements in the DBD source). These can be: packed decimal, zoned decimal, date, time, string, etc; signed or unsigned; and precision and scale. It also uses metadata to analyze an SQL query to convert it into a DL/I call procedure.

The new Virtual Db2 user group is at: itech-ed.com/virtualdb2

Advanced IMS metadata types include maps (DFSMAP) and cases (DFSCASE), which are used for segments whose field layouts are heavily redefined or remapped based on the value of a specific field, for example, an insurance policy segment may have been overloaded to hold different types of policies for vehicle policies, house policies, and endowment policies. They each would have very different field layouts/characteristics. Depending on field, these can now be defined in the same segment or within the same hierarchical path of the database.

Other advanced metadata types are: arrays, which are used to support repeating elements in the database; and user type converters, which are used to define non-standard datatypes. The DBD definition would point to a Java routine that has a readObject and writeObject routine that customers can implement to transform the custom data to a meaningful Java type. This routine is called internally by the IMS Universal drivers.

Joseph Leamon then took over the presentation, where he looked at Universal Drivers. He said that the DLI and JDBC API in the Universal Driver has been

enhanced to use ByteBuffers as input and output areas. The advantages of using this API include: minimal metadata needed to interact with the IMS database; flexible manipulation of data; and the ability to use native storage to share across languages.

Control segments allows users to depend on a key field that exists in a segment that exists above the current segment's hierarchy. The universal drivers utilize the keyfeedback area to determine this key value.

Improvements have been made with hide null field values in the ResultSet when they cannot exist. Cases are used in SQL where clauses will filter out valid and invalid groups in the result set.

There is a special flag to enable SQL WHERE clause optimization. When enabled, the Universal Drivers will analyze the provided SQL statement and optimize when all conditions are met. WHERE clauses with sub-fields that make up a key field are potentially eligible for optimization. Optimized SQL queries will transform into a ranged query that uses the key field with HIGH (FF) and LOW values (00). This will avoid database scans by using a key search whenever possible.

New hex and binary literals can be used in SQL INSERT, UPDATE, SELECT, and DELETE statements.

There's improved array support. The Universal Drivers now support Dynamic Arrays. The IMS Universal JDBC driver allows for easier access to Array datatype fields using the expandArrayResultSet property. This property enables the IMS Universal JDBC driver to create an ArrayResultSet with the array elements as the actual fields in the ResultSet instead of a Struct object containing those elements as the only entry into that ResultSet.

Joseph concluded by saying that, coming soon, metadata consumption has been rearchitected to avoid the need for JAXB. This is highly beneficial for users using higher levels of Java or environments with JAXB dependencies.

A copy of Joseph Leamon and Mahmood Abuzaina's presentation is available for download from the Virtual IMS user group website at: <https://itech-ed.com/virtualims/presentations/IMSJavaDec22.pdf>

You can see and hear the whole user group meeting at: <https://youtu.be/xuAnehPzQro>.

Meeting dates

The following meeting dates have been arranged for the Virtual IMS user group:

- On 7 February, Scott Quillicy, Senior Director, Sales Engineering at Precisely, will be discussing “Lessons Learned – IMS Application Modernization to Cloud”.
- The following meeting will be on 18 April, when Dusty Rivers, Director z Systems Software at Adaptigent will be presenting.

Sponsorship opportunity

Our user group has been in existence since 2007 and

is well-respected among users of IMS. Currently, co-sponsorship is available for the user group. Please contact me (trevor@itech-ed.com).

IMS articles and blogs

Introducing the new IMS Connect ALTPCB Output Enhancement! by Jack Yuan and Sanjay Kaliyur in the IMS part of the IBM Z and LinuxONE Community (28 November 2022). You can find the article at: <https://community.ibm.com/community/user/ibmz-and-linuxone/blogs/sanjay-kaliyur1/2022/11/28/ims-connect-altpcb-output-enhancement>



About the Virtual IMS user group

The Virtual IMS user group was established as a way for individuals using IBM’s IMS hierarchical database and transaction processing systems to exchange information, learn new techniques, and advance their skills with the product

The Web site at <https://itech-ed.com/virtualims> provides a central point for coordinating periodic meetings (which contain technically-oriented topics presented in a webinar format), and provides articles, discussions, links, and other resources of interest to IBM IMS practitioners. Anyone with an interest in IMS is welcome to join the Virtual IMS user group and share in the knowledge exchange.

To share ideas, and for further information, contact trevor@itech-ed.com.

The Virtual IMS user group is free to its members.