DB2 Connections to DB2 for z/OS
“A DB2 Thread End-to-End”
aka: Paul’s Cheatsheet

Paul Wirth
wirthp@us.ibm.com
Disclaimer and Trademarks

Information contained in this material has not been submitted to any formal IBM review and is distributed on "as is" basis without any warranty either expressed or implied. Measurements data have been obtained in laboratory environment. Information in this presentation about IBM's future plans reflect current thinking and is subject to change at IBM's business discretion. You should not rely on such information to make business plans. The use of this information is a customer responsibility.

IBM MAY HAVE PATENTS OR PENDING PATENT APPLICATIONS COVERING SUBJECT MATTER IN THIS DOCUMENT. THE FURNISHING OF THIS DOCUMENT DOES NOT IMPLY GIVING LICENSE TO THESE PATENTS.

TRADEMARKS: THE FOLLOWING TERMS ARE TRADEMARKS OR ® REGISTERED TRADEMARKS OF THE IBM CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: AIX, AS/400, DATABASE 2, DB2, e-business logo, Enterprise Storage Server, ESCON, FICON, OS/390, OS/400, ES/9000, MVS/ESA, Netfinity, RISC, RISC SYSTEM/6000, System i, System p, System x, System z, IBM, Lotus, NOTES, WebSphere, z/Architecture, z/OS, zSeries

The FOLLOWING TERMS ARE TRADEMARKS OR REGISTERED TRADEMARKS OF THE MICROSOFT CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: MICROSOFT, WINDOWS, WINDOWS NT, ODBC, WINDOWS 95

For additional information see ibm.com/legal/copytrade.phtml
Agenda

- Topology of Distributed Applications
- DB2 Clients
- DB2 Connect
- DB2 z/OS and Distributed Connections
- DB2 z/OS and Connection Resources
- z/OS Workload Manager
- DB2 Private Protocol
- Supplemental Materials
Overview:

The purpose of this presentation is to provide an overview of distributed connections to DB2 z/OS. Given distributed transactions cross many different business organizations, gaining an understanding the “big picture” of a distributed transaction is difficult. Frequently I get the comment; “We have a DB2 Connect problem!”. However, the comment is really stating we are have a problem with distributed transactions; and the issue may actually not be with DB2 Connect but some other item. The goal of this presentation is to provide a high level end-to-end view of a distributed transactions and to try and de-mystifying the behavior of DB2 z/OS distributed workloads.
Goal:

DB2 historically had CICS, JES or IMS/TM managing workload and the priority of work; however, in a distributed environment with many different sources of work (workstations, application servers, etc.) DB2 has to be designed to regulate transaction volume and transaction priority. This presentation has the following education goals:

- Configuration of distributed work for resiliency
- Capacity planning of distributed work for DB2
- Workload prioritization
- **Basically, configuring DB2 as a pseudo TP monitor**
5 “Simple” parts to DB2 Connections

1. **Network** - SNA or TCP/IP
2. **DB2 Clients**
   - **Connect** Tools: Control Center, Configuration Assistant, DB2 Command Line Editor & etc…
   - Drivers/Providers:
     - DB2 Call Level Interface (CLI)
     - Microsoft: ODBC, OLE DB and DB2 .NET Data Providers
     - JDBC type 2 and JDBC Type 4 Drivers
3. **DB2 Connect**
   - DB2 Connect PE (This product is specifically designed and licensed for enabling two-tier client/server applications running on individual workstations and, as such, is not appropriate for use on servers.)
   - DB2 Connect Server (Same code – different licenses)
     - DB2 Connect Enterprise Edition - (Purchased by No. of Users or No. of Concurrent Connections)
     - DB2 Connect Unlimited Edition for zSeries (MSU – Includes DB2 Connect PE)
     - DB2 Connect Unlimited Edition for 1Series (includes DB2 Connect PE)
   - JDBC Type 4 Driver  (A client that can connect directly to DB2 for z/OS with DB2 Connect license) 8.1 FP10+
     (Note: Version 9.1 fix pack 5 and Version 9.5 fix pack 1 are the recommended minimum levels)
   - IBM Data Server Driver (CLI, ODB, OLE-DB or ADO.Net with DB2 Connect license) 9.5 FP3+
4. **DB2 z/OS**
   - DDF – Network Connections - CONDBAT
   - DBM1 – Database Access Threads – DBATs
5. **z/OS Workload Manager (WLM)** - Service Classes and Classification Rules
Typical Topologies for Application Connections to DB2 z/OS

- This example has DB2 Connect Enterprise Server Edition, being use as a gateway.
- Optionally, DB2 Connect Personal Edition allows client to connect directly to DB2 z/OS.
- Optionally, DB2 Connect using Java Universal driver TYPE-4 allows Java network connects to connect directly to DB2 z/OS.
- DB2 9.5 fix pack three provide IBM Data Server Driver which can also connect directly to DB2 z/OS too.
- DB2 SYSPLEX used for scalability and high availability.
Summary of DB2 Connect Topologies - Roadmap:

1. Web Application server with DB2 Connect or Type-4 Java Driver or IBM Data Server Driver

2. **Web Application server With DB2 Connect or Type-4 Java Driver or IBM Data Server Driver**

3. DB2 Connect PE

4. **DB2 Connect Gateway server on zLinux and Hypersockets**

**Tip:** When agents are on the same computer as DB2 Connect set the following DB2 registry variable: db2connect_in_app_process=n

**Tip:** Hypersockets are used for DB2 sub systems local to the CEC. Remote DB2s use normal TCP/IP
Things to consider for: availability, redundancy and resiliency in DB2 Connect Gateway Topologies:

1. Edge server
   - Web Application server with DB2 Connect or Type-4 Java Driver or IBM Data Server Driver
   - Excel IBM Client
   - Web Application server IBM Client

2. IP Sprayer (Load Balancing)
   - DB2 Connect Gateway server on zLinux and Hypersockets
   - Edge server
   - IP Sprayer (Load Balancing)

3. zIIP DB2 - DDF - DBM1 WLM
   - DB2 Connect Gateway server on zLinux and Hypersockets

***Tip:*** When setting up a high-availability system you want to eliminate single points of failure. A potential single point of failure is your network controller. You can use Virtual IP Addressing (VIPA) instead of using a particular network controller. In case of a network controller failure, another controller can be used.
Things to consider for: availability, redundancy and resiliency in DB2 Connect Gateway Topologies…:

1. Installing DB2 Connect/Data Server Driver/Java Type-4 on the application server avoids the additional network hop through a DB2 Connect gateway and avoids the DB2 Connect gateway as a single point of failure. (SAP does this for their DBs on DB2 z/OS)

2. Using DB2 Connect gateways centralizes thread pool management and gateway software maintenance. Use IP Sprayers to avoid DB2 Connect gateway as single point of failure. Note: make sure and rigorously test your IP Sprayer solution because there are lots of combinations available and some work better than others.

3. DB2 Connect on zLinux exploits hypersockets for local DB2 systems in the same CEC. Note: Think about redundancy of DB2 Connect even on zLinux…

- Use Dynamic VIPA for network component transparency for DB2 data sharing
- Use Sysplex Distributor for application transparency (DB2 data sharing)
- Life is not “either or other” do both – Try using DB2 Connect/Data Server Driver/Type-4 driver on application servers and DB2 Connect gateways for end-user connections

- Be careful when considering the elimination of DB2 Connect Gateways that you don’t violate current licensing agreement with Data Server Drivers
New Changes IBM Data Server Drivers

- The IBM Data Server Driver for JDBC and SQLJ is capable of performing sysplex workload balancing functions since Version 8.1 fix pack 10. Because improvements have been made and support has been added, Version 9.1 fix pack 5 and Version 9.5 fix pack 1 are the recommended minimum levels.

- Non-Java data server drivers, starting with Version 9.5 fix pack 3 that have a DB2 Connect license can also access a DB2 for z/OS Sysplex directly without going through a middle-tier DB2 Connect Server.

- With the non-Java drivers shipped with Version 9.5 fix pack 3, Sysplex Workload Balancing (WLB) is available on commit boundaries even on a single connection.

- If you are connecting to a DB2 for z/OS, DB2 for i5/OS, or DB2 for VSE and VM server with the IBM Data Server Clients or Driver including the JDBC Type4 driver, you must have the appropriate DB2 Connect license.

- Multi-transport models such as Tuxedo, WebLogic, and Encina still require a middle-tier DB2 Connect Server.
Running DB2 Connect on Application Servers

The DB2 registry variable DB2CONNECT_IN_APP_PROCESS allows applications running on the same machine as a DB2 Connect server product to either have DB2 Connect run within the applications process, default behavior, or to have the application connect to the DB2 Connect server product and then have the host connection run within an agent. For an application to use connection pooling or connection concentration, the connections to the host must be made from within the DB2 Connect server product agents and thus DB2CONNECT_IN_APP_PROCESS must be set to NO (db2connect_in_app_process=n).

Since local connections will appear to DB2 Connect as though they were remote connections, you can use the LIST DCS APPLICATIONS and GET SNAPSHOT FOR ALL DCS APPLICATIONS.

Note: SAP uses this configuration where DB2 Connect is installed on the application server with their application servers to simplify network setup and improve performance.
Agenda

- Topology of Distributed Applications
- DB2 Clients
- DB2 Connect
- DB2 z/OS and Distributed Connections
- DB2 z/OS and Connection Resources
- z/OS Workload Manager
- DB2 Private Protocol
- Supplemental Materials:
What are we talking about on the “Roadmap”?

1. Web Application server with DB2 Connect or Type-4 Java Driver or IBM Data Server Driver

2. IP Sprayer (Load Balancing)

3. Virtual IP Addressing (VIPA) & SysPlex Distributor
Some DB2 Client History

- In DB2 8, the client communication protocol was changed from the proprietary DB2RA protocol to the open standard Distributed Relational Database Architecture (DRDA). This created a streamlined communication infrastructure for the IBM data server family, as well as reduced the code path length and CPU load on clients, which typically results in better performance.

- In DB2 9.5, access was further simplified access to DB2 for i5/OS and DB2 for z/OS data servers by building on this converged code base.
Which IBM client connectivity option is right for you?

The evolution of the DB2 client connectivity options in the last couple of DB2 releases and new "NAMEs"

<table>
<thead>
<tr>
<th>DB2 8 and 8.2 Clients End of Service 30 April 2009</th>
<th>DB2 9 Clients (Clients and Drivers)</th>
<th>DB2 9.5 Clients</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Administration Client</td>
<td>DB2 Client</td>
<td>IBM Data Server Client</td>
</tr>
<tr>
<td>DB2 Runtime Client</td>
<td>DB2 Runtime Client</td>
<td>IBM Data Server Runtime Client</td>
</tr>
<tr>
<td>Java Common Client 3.0</td>
<td>IBM DB2 Driver for 3.0 JDBC and SQLJ</td>
<td>IBM DB2 Driver for 3.0 and 4.0 JDBC and SQLJ</td>
</tr>
<tr>
<td>DB2 Application Development Client</td>
<td>IBM Data Studio</td>
<td>IBM Data Studio</td>
</tr>
<tr>
<td>DB2 LUW, iSeries, DB2 VM/VSE DB2 on z/OS</td>
<td>DB2 LUW, iSeries, DB2 VM/VSE DB2 on z/OS</td>
<td>DB2 LUW, iSeries, DB2 VM/VSE DB2 on z/OS, Informix and U2</td>
</tr>
<tr>
<td>DB2 9 for z/OS Min. DB2 8 FP13</td>
<td>DB2 9 for z/OS Min. DB2 9 FP1</td>
<td></td>
</tr>
</tbody>
</table>

Clients and Drivers Concept: Clients big (100MB+) and Drivers small (less than 30MB)
DB2 (LUW) 9.1

Note: DB2 9 FP2 added support for Windows Vista, this is described in our Support for Microsoft Windows Vista page. http://www-01.ibm.com/support/docview.wss?uid=swg21249760

- The **DB2 9 Runtime Client** is best option if your only requirements are to enable applications to access DB2 9 data servers. They provide the APIs necessary to perform this task, but this client comes with no management tools.

- The **DB2 9 Client** includes all the functions found in the DB2 Runtime Client plus functions for client-server configuration, database administration, and application development through a set of rich graphical tools. The DB2 9 Client replaces the functions found in both the DB2 8 Application Development and DB2 8 Administration clients and IBM .Net Add-Ins.

- **IBM DB2 Driver for ODBC and CLI** is a 13MB application programming interface (API) and the ODBC API. Though the DB2 Client and DB2 Runtime Client both support the DB2 CLI and ODBC APIs, this driver is not a part of either DB2 client. It is available separately, installed separately, and supports a subset of the functionality of the DB2 clients.

- **IBM DB2 Driver for JDBC and SQLJ (Aka the Java Common Client (JCC)/Universal Driver/Type-4 driver)** is a 9 MB fully redistributable client provides JDBC and SQLJ applications access to DB2 data servers without installing and maintaining DB2 client code. If you are connecting to a DB2 for System i or DB2 for System z data server, you are still required to purchase the DB2 Connect product.

(Note: The Application Development Client is gone. Development Center has been replaced by IBM Data Studio.)
DB2 (LUW) 9.5

**Note: “IBM Data Server” name 9.5 Includes Informix support now**

- **The IBM Data Server Runtime Client:** provides a means for applications to be executed against remote DB2 databases. GUI tools are not shipped with the IBM Data Server Runtime Client. Support for common database access interfaces: JDBC, ADO.NET, OLE DB, ODBC, DB2 Command Line Interface (CLI), PHP, and Ruby. This includes drivers and capabilities to define data sources. IBM Informix Dynamic Server support for PHP, Ruby, .NET and JDBC.

- **The IBM Data Server Client:** is a collection of client application drivers and tools for DB2 and Informix IDS data servers. The set of application drivers for DB2 include: embedded SQL, ODBC/CLI, JDBC/SQLJ, OLEDB, .NET, PHP, Perl, and Ruby. Data access and administration tools are providing including: DB2 Control Center (Windows and Linux) and DB2 Command Line Processor (CLP). The set of application drivers for Informix IDS v11.10, or later, include: JDBC, .NET, PHP, and Ruby. The IBM Database add-ins for Visual Studio 2005 are provided to simplify data access for .NET applications for both DB2 and IDS data servers.

- **IBM Data Server Driver for ODBC, CLI:** is a 13MB lightweight deployment solution for Windows applications. It provides runtime support for applications using DB2 CLI API, ODBC API, or .NET API without the need of installing the Data Server Client or the Data Server Runtime Client. IBM Informix Dynamic Server support for .NET, PHP, and Ruby.

- **IBM DB2 Driver for JDBC and SQLJ (Aka the Java Common Client (JCC)/Universal Driver/Type-4 driver):** is a 9 MB fully redistributable client provides JDBC and SQLJ applications access to DB2 data servers. This 2 MB fully redistributable client provides JDBC and SQLJ applications access to DB2 data servers without installing and maintaining DB2 client code. If you are connecting to a DB2 for System i or DB2 for System z data server, you are still required to purchase the DB2 Connect product. (fix pack 1 adds JDBC 4.0 Support)
DB2 (LUW) 9.5 (Continued)

Note: “IBM Data Server” name 9.5 Includes Informix support now

- **IBM Data Server Driver Package**: is a 25MB lightweight deployment solution for Windows applications. For applications using ODBC, CLI, .NET, OLE DB, PHP, Ruby, JDBC, or SQLJ, use IBM Informix Dynamic Server support for .NET, PHP, and Ruby.
  - PHP and Ruby drivers. These header files are also available starting in Version 9.5 Fix Pack 3.
    - Note: Python and Perl drivers are not available in IBM Data Server Driver Package - however, you can download and build these drivers using the application header files that are included
  - DB2 Interactive Call Level Interface (db2cli) support. This support is also available starting in Version 9.5 Fix Pack 4
  - DRDA® traces (db2drdat) support. This support is also available starting in Version 9.5 Fix Pack 4.
  - Combines **IBM Data Server Driver for ODBC, CLI and IBM DB2 Driver for JDBC** plus .Net
DB2 (LUW) 9.7

- **The IBM Data Server Runtime Client:** provides a means for applications to be executed against remote DB2 databases. GUI tools are not shipped with the IBM Data Server Runtime Client. Support for common database access interfaces: JDBC, ADO.NET, OLE DB, ODBC, DB2 Command Line Interface (CLI), PHP, and Ruby. This includes drivers and capabilities to define data sources. IBM Informix Dynamic Server support for PHP, Ruby, .NET and JDBC.

- **The IBM Data Server Client:** is a collection of client application drivers and tools for DB2 and Informix IDS data servers. The set of application drivers for DB2 include: embedded SQL, ODBC/CLI, JDBC/SQJ, OLEDB, .NET, PHP, Perl, and Ruby. Data access and administration tools are providing including: DB2 Control Center (Windows and Linux) and DB2 Command Line Processor (CLP). The set of application drivers for Informix IDS v11.10, or later, include: JDBC, .NET, PHP, and Ruby. The IBM Database add-ins for Visual Studio 2005 are provided to simplify data access for .NET applications for both DB2 and IDS data servers.

- **IBM Data Server Driver for ODBC, CLI:** is a 13MB lightweight deployment solution for Windows applications. It provides runtime support for applications using DB2 CLI API, ODBC API, or .NET API without the need of installing the Data Server Client or the Data Server Runtime Client. IBM Informix Dynamic Server support for .NET, PHP, and Ruby.

- **IBM DB2 Driver for JDBC and SQLJ (Aka the Java Common Client (JCC)/Universal Driver/Type-4 driver):** is a 9 MB fully redistributable client provides JDBC and SQLJ applications access to DB2 data servers. This 2 MB fully redistributable client provides JDBC and SQLJ applications access to DB2 data servers without installing and maintaining DB2 client code. If you are connecting to a DB2 for System i or DB2 for System z data server, you are still required to purchase the DB2 Connect product.

- **IBM Data Server Driver Package:** is a 25MB lightweight deployment solution for Windows applications. For applications using ODBC, CLI, .NET, OLE DB, PHP, Ruby, JDBC, or SQLJ, use . IBM Informix Dynamic Server support for .NET, PHP, and Ruby. Combines IBM Data Server Driver for ODBC, CLI and IBM DB2 Driver for JDBC plus .Net DB2® Command Line Processor Plus (CLPPlus) for dynamically creating, editing, and running SQL statements and scripts.
So what would I see at the download site…
DB2 9.7 fix pack 1
Current Client Names
Choosing the right Client/Driver

- There is a functional overlap
- Should balance functionality with footprint

IBM Data Server Client: Includes Developer and DBA GUI tools

IBM Data Server Runtime Client: Includes CLP

IBM Data Server Driver Package: Includes support for .NET, OLE DB, PHP, Ruby, Perl

- IBM Data Server Driver for ODBC and CLI
- IBM Data Server Driver for JDBC and SQLJ
Client Deployment Strategies

- Majority of client workstations are runtime oriented
  - End user workstations
  - App servers
- Runtime deployment
  - IBM Data Server Driver highly recommended - Lightest runtime focused client package
  - Common client configuration across multiple levels
  - No migration to move to newer levels - No database directory
- Developer deployment
  - Essentially runtime plus tooling
  - Start with IBM Data Server Driver
  - Add Tooling package:
    - Data Studio for Java
    - Microsoft Visual Studio Add-Ins for .NET
- Administrator deployment
  - Using CLP for administration - Investigate use of CLP Plus
  - Use Data Server Runtime Client
- DBA and All others
  - Use Data Server Client
- Reminder
  - The larger the client package, the more that is included, the more that needs to be rebound
Cool places to review DB2 Client options and tips…

- Which DB2 9.5 client connectivity option is right for you?  

- Some new DB2 for z/OS options:
  - In DB2 9.5, access was further simplified to DB2 for i5/OS and DB2 for z/OS data servers by building on this converged code base.
  - As of DB2 9.5, you can enable DB2 Connect Personal Edition (DB2 Connect PE) on any DB2 data server, IBM Data Server **Runtime Client**, or an IBM Data Server **Client** by simply adding a valid DB2 Connect PE license using the DB2 License Center or the db2licm command. In DB2 9, you actually would have to install DB2 Connect PE on a workstation to enable this functionality.
  - This enhancement is terrific for deployments, especially development environments where DB2 Connect PE is typically found. Now you can enable DB2 for i5/OS or DB2 for z/OS development by simply adding a license to a DB2 9.5 installation.

- Client landing page  
  - Easy access to latest drivers
  - Navigation to older driver levels
    - Still being built up
Client-side support for DB2 for z/OS® Sysplex capabilities in IBM data server clients and non-Java data server drivers that have a DB2 Connect license.

**DB2 Connect Sysplex support**

A Sysplex is a collection of System z® servers that cooperate, using hardware and software, to process work. The Sysplex coordinates the cooperation by increasing the number of processors working together, which increases the amount of work that can be processed. In addition to an increase in processing capability, a Sysplex can provide flexibility in mixing levels of hardware and software, and in dynamically adding systems.

Sysplex permits DB2® Connect™ to seamlessly balance connections across different members of a data sharing group. Sysplex also provides DB2 Connect the means to try alternate members should a failure occur with one member. The rerouting capability for Sysplex is a DB2 Connect feature. DB2 Connect support for Sysplex is enabled by default and so is the rerouting capability for Sysplex. Sysplex support to a host database can be turned off by removing the SYSPLEX parameter from its DCS directory entry, but the DCS entry itself should not be removed, even if it has no other parameter specified.

With the automatic client reroute capability for Sysplex, the default behavior is for a Sysplex enabled connection to retry on the connect when there is a communication failure. Special register values, up until the last successful transaction not holding resources, are replayed when DB2 Connect is connected to a DB2® for z/OS® server.

You can configure the exact automatic client reroute retry behavior, including disablement, by using the DB2_MAX_CLIENT_CONNRETRIES and DB2_CONNRETRIES_INTERVAL registry variables. The connection timeout registry variable is DB2TCP_CLIENT_CONNTIMEOUT.
IBM Data Server **Drivers** Provides Improvements for Microsoft Customers DB2 9.5  
(Basically Type-4 driver for non-Java)

- A new configuration file, `db2dsdriver.cfg`, contains database directory information and **client configuration parameters in a human-readable format** that can be used to configure the behavior of ODBC, CLI, OLE-DB and .NET or open source (Perl, PHP) and the applications using the keywords.
- Not for use with the Java driver
- This configuration file can be used with the following:
  - Drivers:
    - IBM Data Server Driver for ODBC and CLI,
    - IBM Data Server Driver Package
  - Clients:
    - IBM Data Server Client
    - IBM Data Server Runtime Client
- **Tip:** Syntax errors are silently ignored in `db2dsdriver.cfg`. To ensure your settings are used, update your database configuration using `diaglevel` 4 through CLP (or manually update `db2cli.ini` if using the thin data server drivers where CLP is not available) and check `db2diag.log` for error messages.
- DB2 9.7 fix pack 1 includes DB2 Command Line Processor Plus (CLPPlus) for dynamically creating, editing, and running SQL statements and scripts
- Note: For IBM data server clients, the `db2dsdriver.cfg` configuration file is used only to retrieve Sysplex-related settings, such as WLB and ACR. Database connection information and properties should come from a different source, such as the database catalog, a connection string, the `db2cli.ini` initialization file, or .NET object properties.
- **Also future fix pack “maybe DB2 9.7 fix pack 2” IBM Data Server Driver will provide a command line utility to register ODBC DSN on Windows ODBC Driver Manager**
IBM and Continues Improvement for Microsoft .Net…

- If the failure occurs on the first SQL operation in a transaction, and seamless failover is enabled (and applicable), the client can replay the failed SQL operation as part of **Automatic Client Reroute (ACR)** processing. If the connection is successful, no error is reported to the application.
- If ACR is enabled, an application that encounters a connectivity failure is routed to another member of the Sysplex. If seamless failover is disabled or not applicable, the error SQL30108N is returned to the application.
- To perform ACR, the client must use a TCP/IP connection and have a DB2® Connect™ license. The following Version 9.5 Fix Pack 3 or later clients provide support for ACR:
  - IBM® Data Server Client, IBM Data Server Runtime Client, IBM Data Server Driver Package
- Automatic client reroute settings in the *db2dsdriver.cfg* configuration file
- **enableAcr**: Specifies whether the ACR feature is on (true) or off (false). This setting defaults to true when transaction-level load balancing is enabled (the enableWLB setting is true). If ACR is disabled, any other settings for ACR are ignored.
- **enableSeamlessAcr**: Specifies whether the ACR with seamless failover feature is on (true) or off (false). If ACR is enabled and the target server is DB2 for z/OS, seamless ACR is enabled by default. If the target server is not DB2 for z/OS, this setting is ignored, and seamless ACR is not supported.
SQL Packages

- Dynamic SQL packages standardized between CLI, .NET and JCC in V8
- Only changed once since then (driven by z/OS unique change)
- Changes documented:
  http://www.ibm.com/developerworks/wikis/display/DB2/DB2+and+.NET+FAQs#DB2
  and.NETFAQs-ChangesinCLI%2F.NETandJDBCpackages

- Changes in CLI / .NET and JDBC packages
  - The packages were standardized in V8 of the DB2 LUW and DB2 Connect clients. The following is the change history of these packages:
  - Dynamic packages SYSSHxyy, SYSSNxyy, SYSLHxyy, SYSLNxyy:
    - V8 GA - created
  - SYSSTAT misc static statements:
    - V8 GA - created
    - V9.1 FP2, V9.5 GA - added additional sections for Jcc z/OS type 2
- Rebinds only need to happen when upgrading across one of the change points listed above. For example, upgrading from V8 GA to V9.1 GA would not require a rebind. However, any upgrade from V9.1 FP1 would technically require a rebind because the SYSSTAT package changed. However, the change affected the Jcc z/OS type 2 driver only, thus CLI and .NET, and other servers, need not rebind as they will be unaffected by the change.
DB2 Clients Provide

- Support for common database access interfaces: **JDBC, ADO.NET, OLE DB, ODBC, and DB2 Command Line Interface (CLI)**. This includes drivers and capabilities to define data sources. For example, for ODBC, installing a DB2 client installs the DB2 ODBC driver and registers the driver. Application developers and other users can use the Windows ODBC Data Source Administrator tool to define data sources.
- Base client support to handle database connections, SQL statements, XQuery statements and DB2 commands.

- LDAP exploitation.

- Support for common network communication protocols: TCP/IP, Named Pipe.

- Versions that run on 32-bit and 64-bit operating systems.

- License terms that allow free redistribution of DB2 Runtime Client with your application.
The DB2 V9 Providers/Drivers

1. **DB2 .NET Data Provider**: A high performance, managed ADO.NET data provider. This is the recommended* .NET data provider for use with DB2 family databases. ADO.NET database access using the DB2 .NET Data Provider has fewer restrictions, and provides significantly better performance than the OLE DB and ODBC .NET bridge providers.

2. **OLE DB .NET Data Provider**: A bridge provider that feeds ADO.NET requests to the IBM OLE DB Provider (by way of the COM interop module). This .NET data provider is not recommended* for access to DB2 family databases. The DB2 .NET Data Provider is faster and more feature-rich.

3. **ODBC .NET Data Provider**: A bridge provider that feeds ADO.NET requests to the IBM ODBC Driver. This .NET data provider is not recommended* for access to DB2 family databases. The DB2 .NET Data Provider is faster and more feature-rich.

4. **JDBC type 2** driver are translated to Java native methods. The Java applications that use this driver must run on a DB2 client, through which JDBC requests flow to the DB2 server. DB2 Connect™ must be installed before the DB2 JDBC application driver can be used to access DB2 UDB for iSeries data sources or data sources in the DB2 for OS/390® or z/OS® environments.

5. **JDBC Type 4** JDBC driver is a two-tier pure Java(TM) JDBC driver that allows a Java client to communicate directly with DB2 servers via the DRDA(R) protocol.

6. **DB2 Call Level Interface (CLI)** is IBM's callable SQL interface to the DB2 family of database servers. It is a 'C' and 'C++' application programming interface for relational database access that uses function calls to pass dynamic SQL statements as function arguments. It is an alternative to embedded dynamic SQL, but unlike embedded SQL, DB2 CLI does not require host variables or a precompiler. DB2 CLI is based on the Microsoft(R)** Open Database Connectivity** (ODBC) specification, and the International Standard for SQL/CLI.

*Source: DB2 Version 9 for Linux, UNIX, and Windows - Developing ADO.NET and OLE DB Applications, Developing ADO.NET and OLE DB Applications
IBM DB2 Driver for ODBC, CLI

Installation/Configuration:

- Unzip file for ODBC & CLI version or Execute install file for ODBC, CLI and .Net
- Add valid DB2 Connect license key
- To db2cli.ini ..
  - [ZOS_DB]
  - database=ndcddb202
  - protocol=tcpip
  - hostname=9.39.64.151
  - servicename=446
- Run `db2oreg1.exe -i` to add driver to Windows Registry

Now for the gotches: `db2oreg1.exe` only registers the driver and not the data source. Therefore, programs like Excel will not see the data source (ZOS_DB) unless it is manually added using the ODBC Admin. Tool.

If DB2 databases need to be automatically included as data sources then you will need to use IBM Data Server Runtime Client or IBM Data Server Client. If you want a replacement of the DB2 Runtime Lite Client consider using `db2iprune` with IBM Data Server Runtime Client.
Java Universal Driver

- Java Universal Driver provided in DB2 V8.1 LUW fix pack 4+ called “IBM Data Server Driver for JDBC and SQLJ“
  - Archive Class file: **db2jcc.jar 3.0**
    - Note: new **db2jcc4.jar** 4.0 IBM Data Server (9.5 FP1)
  - JDK 1.3.1 or later 4
  - JDBC 3.0 features require a minimum level of JDK 4 1.4
  - Provides Type-2 and Type-4 Connections
  - fix pack 10 provided Data Sharing Sysplex support for Type-4 Connection to DB2 z/OS, recommended minimum fix pack level 13
  - Java Package name: **com.ibm.db2.jcc** Contains the DB2-specific implementation of JDBC for the DB2 Universal JDBC driver.

- Old Java Driver (pre-universal Driver) is called “DB2 JDBC Type 2 Driver” aka “Legacy” driver
  - Archive Class file: **db2java.zip**
  - Java Package name: **COM.ibm.db2.jdbc** Contains the DB2-specific implementation of the JDBC for the DB2 JDBC Type 2 Driver.
  - The CLI-based Type 2 and Type 3 JDBC drivers are deprecated. No new features or enhancements will be put into these drivers and they will not be available in future DB2 releases. The completely redesigned Universal JDBC driver is provided to replace these legacy drivers with more enhanced features. You are encouraged to migrate applications to use the new driver as soon as possible. DB2 LUW 8.2 Application Development Guide: Building and Running Applications.
  - **The Legacy driver will not be supported by DB2 9 for z/OS**
IBM Data Server Universal Java Type 4 Driver files

Type 4 drivers are pure Java and implement the network protocol for a specific data source. The client connects directly to the data source.

- db2jcc.jar – Type 4 driver JDBC 3.0 and earlier
- db2jcc4.jar – Type 4 driver JDBC 4.0 and earlier (DB2 9.5 FP1+)
- sqlj.zip – SQLJ Support JDBC 3.0 and earlier
- sqlj4.zip – SQLJ Support JDBC 4.0 and earlier (DB2 9.5 FP1+)
- db2jcc_license_cisuz.jar - License for : Cloudscape All DB2 Database for Linux, UNIX, and Windows servers DB2 for z/OS DB2 UDB for iSeries (Included with DB2 Connect)

How to find IBM DB2 Driver for JDBC and SQLJ version and environment information:

```
java -com.ibm.db2.jcc.DB2Jcc -version -configuration help
```
IBM Data Server Driver and db2dsdriver.cfg

- ODBC, CLI, OLE-DB and .NET or open source (Perl, PHP)
- No database directory
- The db2dsdriver.cfg file is XML based,
- A schema file – dsdriver.xsd, is shipped to allow client side validation of the file
- The db2dsdcfgfill command copies the existing database directory information from either the existing IBM Data Server Client or IBM Data Server Runtime Client, and copies the information to the db2dsdriver.cfg configuration file
- No migration to move to newer levels because there is no database directory
Example of a db2dsdriver.cfg file and enable ACR for the database SAMPLE
Note: The db2dsdriver file is an XML file

```xml
<database name="SAMPLE" host="v33ec065.my.domain.com" port="446">
  <!-- database-specific parameters -->
  <WLB>
    <!-- Sysplex WLB is disabled by default -->
    <parameter name="enableWLB" value="true" />
  </WLB>
  <ACR>
    <!-- ACR is already enabled by default when WLB is enabled -->
    <parameter name="enableAcr" value="true" />
  </ACR>
</database>
```
Summary of DB2 Connections

1. **DB2 Connect PE**
   - DB2 Connect/PE:
     - DNS: my.sysz.com
     - Port: 446
     - Location Name: SYSZDB2T

2. **Web Application server With DB2 Connect or Type-4 Java Driver or IBM Data Server Driver**

3. **DB2 z/OS**
   - Subsystem Name: DB2T
   - DNS: my.sysz.com
   - Port: 446
   - Location Name: SYSZDB2T

- IBM Client
- Internet Explorer
- Excel
- IBM Data Server Client
- Gateway servers
- DB2 Client:
  - DNS: my.db2conn.com
  - Port: 50000
  - Location Name: SYSZDB2T
- Application server
Examples of manually cataloging client and gateway connections – Using DB2 Client Catalog commands

DB2 client catalog statements:
- catalog tcpip node ndcdb205 remote my.db2conn.com server 50000 ostype os390
- catalog db ndcdb205 at node ndcdb205

DB2 gateway catalog statements:
- catalog tcpip node ndcdb205 remote my.sysz.com server 446 ostype os390
- catalog db ndcdb205 at node ndcdb205
- catalog dcs database ndcdb205 as ndcdb205
  (Note: the dcs connection identifies this as a “host” connection)
Catalog commands:

- **Node** entry:
  - Nick name for: communication protocol, and network location

- **DB** entry:
  - Database name and alias database name if two database have the same name
  - Authentication - SERVER, SERVER_ENCRYPT, CLIENT, KERBEROS, and DATA_ENCRYPT

- **DCS (aka host)** entry:
  - Identify remote database as z/OS, OS/400, VSE or VM
  - Additional parameters:
    - SQLCODE map-file, disconnect processing, INTERRUPT_ENABLED, **SYSPLEX**, LOCALDATE format, Bidirectional CCSID
DB2 z/OS V8 Server LOCATION Aliases

- BSDS for each DB2 member can include multiple LOCATION names for the DB2 server
- Each LOCATION name is registered with WLM separately for WLM sysplex workload balancing information
- Useful for:
  - consolidating DB2 systems
  - workload routing within sysplex

The extra location alias names are defined in the BSDS with the Change Log Inventory utility (DSNJU003).
Additional Connection Information DB2 z/OS

- A DB2 Universal Database for z/OS and OS/390 subsystem identified by its LOCATION NAME or one of the alias LOCATION names defined on the z/OS server. The LOCATION NAME can be determined by the DSNL004I message (LOCATION=location), which is written when the Distributed Data Facility (DDF) is started.

  The -DISplay DDF command could also be used.

- If accessing a z/OS data sharing group, the domain name should map to the DB2 group dynamic VIPA address. This address routes to the least loaded DB2 member. To access a specific member use the specific DB2 member dynamic VIPA address and turn off sysplex routing. Each member DSNL004I message displays the member specific domain name.

- The 6th positional parameter on the DCS entry, can be used to explicitly enable DB2 Connect SYSPLEX support for a particular database.
Configuration Assistant - GUI to setup connections

In addition to configuring database and testing connections, connection profiles can be created too. Connection profiles are useful for replication of client setup.

Test connections with any of the drivers…
What does a Java Type-4 connection look like? (Data Studio includes a Type-4 driver no client is needed)
DB2 9 for z/OS Supports SSL from Java Type-4 Driver

- The IBM® Data Server Driver for JDBC and SQLJ provides support for the Secure Sockets Layer (SSL) through the Java™ Secure Socket Extension (JSSE).
- You can use SSL support in your Java applications if you use IBM Data Server Driver for JDBC and SQLJ type 4 connectivity to DB2® for z/OS® Version 7 or later, to DB2 Database for Linux®, UNIX®, and Windows® Version 9.1, **Fix Pack 2 or later**, or to IBM Informix® Dynamic Server (IDS) Version 11.50 or later.
- To configure database connections under the IBM® Data Server Driver for JDBC and SQLJ to use SSL, you need to set the DB2BaseDataSource.sslConnection property to true.
- **Prerequisite:** Before a connection to a data source can use SSL, the port to which the application connects must be configured in the database server as the SSL listener port.
- You can set DB2BaseDataSource.sslConnection on a Connection or DataSource instance. The following example demonstrates how to set the sslConnection property on a Connection instance:

```
java.util.Properties properties = new java.util.Properties();
properties.put("user", "xxxx");
properties.put("password", "yyyy");
properties.put("sslConnection", "true");
java.sql.Connection con = java.sql.DriverManager.getConnection(url, properties);
```

- See DB2 9.5 Information Center for additional details to setup the Java runtime for SSL
DB2 9 for z/OS Supports SSL in non-Java clients (9.1 fix pack 5)

- What's new for V9.1 FP5: Secure Sockets Layer (SSL) protocol is supported by the database client
  - Starting with Fix Pack 5, SSL is supported on the non-Java DB2® clients. Prior to Fix Pack 5, SSL was supported for Java clients only.
  - Secure Sockets Layer (SSL) support ensures that the transfer of sensitive information over the network is secure. SSL is now supported on all Version 9.1 DB2 clients and on the DB2 Version 9.1 server.
- See DB2 9.5 Information Center for additional details to setup the Java runtime for SSL
Visual Studio 2005/8 Enter Connection and Filtering Information

The Steps below are used to select connection information using connection information:

- After DB2 .Net Add-Ins are registered, the IBM (DB2) Data Provider will be listed in the Data sources list box. To select other database press the “change” button. DB2 is listed first because we want to be first ☺

- Enter the TCP/IP address or DNS name and TCP/IP port: demomvs.demopkg.ibm.com:4462

- Enter User ID and Password
  (Save my password is optional)
  For the demo the password was saved to minimize steps.

- Enter database or location name: NDCDB201

- Press the “Specify filtering options” button to add filtering schema/owner name to reduce objects in output lists.

- Notice the Schema filter is including objects from two schemas: “DBA031 and DBA015”.

- The Object Name Filter example shows how to filter tables that begin with the letter “D”.
So how do these driver/provider things work?  
“It’s as easy as 1, 2, 3…”

Example:  
- Excel Using  
- DB2 Client  
- ODBC (Driver/Provider)
I am having problems connecting from Microsoft Visual Studio…


- There are many potential causes for connectivity issues, to aid in debugging, IBM has provided the `testconn` utility, installed with DB2 9. Testconn is a small .NET application that directly uses the DB2 .NET data provider and performs a test connection and some simple operations to validate server setup.
- There are two versions `testconn11` for FW 1.1 and `testconn20` for FW 2.0. Both have the same syntax, but they will load the proper framework version of the DB2 .NET provider. For invocation, simply provide a .NET connection string:
  ```
testconn20 database=sample;server=myhost:50000;userid=myid;password=mypass
  ```
- A complete list of connection string keywords is available in the `ConnectionString` property online help.
- If testconn encounters any errors, it will display them along with suggested resolution steps. Here is a sample testconn20 output:
  - Step 1: Printing version info
    `.NET Framework version: 2.0.50727.42
    DB2 .NET provider version: 9.0.0.2
    Capability bits: ALLDEFINED
    Build: 20060813
    Elapsed: 0.150591`
  - Step 2: Connecting using "database=sample"
    `DB2 Server type and version: DB2/NT 09.01.0000
    Elapsed: 0.9336642`
  - Step 3: Selecting rows from SYSIBM.SYSTABLES to validate existence of packages
    `SELECT * FROM SYSIBM.SYSTABLES FETCH FIRST 5 rows only
    Elapsed: 0.050197`
  - Step 4: Calling GetSchema for tables to validate existence of schema functions
    `Elapsed: 0.2208668`
  - Test passed.
Updating db2cli.ini

1. Open DB2 Configuration Assistant
2. Right click on desired DB2 z/OS
3. Select “CLI Settings…”
4. Select “Settings” tab
5. Select “Add”
6. Select desired parameter and adjust setting

NOTE: Avoid editing db2cli.ini manually, parameters do have requirements of position in the file, use DB2 Configuration Assistant to avoid mistakes
A brief overview of optional information on DB2 Client tools…
“Thin Client”

- This is an alternate method for installing an IBM data server client that leverages the Windows support for thin client topologies. This method can be used to install the DB2 Administration Client(8.x), IBM Data Server Client(9.5) or DB2 Connect Personal Edition on Windows. This method does not apply to the IBM Data Server Runtime Client nor IBM Data Server river for ODBC, CLI, and .NET.

- A thin client topology or thin client topology environment consists of one thin client code server and one or more thin clients. The IBM data server client code is installed on the code server, rather than on each client workstation. On each thin client workstation, only minimal amount of code and configuration is required. When a thin client initiates a database connection, IBM data server client code is dynamically loaded from the code server as required. The thin client then connects to the database in the usual fashion.
“Thin Client” continued…

- By implementing this type of environment, disk space requirements for each thin workstation are reduced (approximately 16–112 MB per workstation can be saved), and the code only needs to be installed, updated, or migrated on one machine.

- Potential loss of system performance at program initialization because programs must load from a code server across a LAN connection. Time extent of performance loss will depend on variables such as the load and speed of both the network and the code server.

- Setting up a thin client environment involves setup on both the code server and each thin client workstation. Steps 1 to 3 are performed on the code server machine and the remaining steps are performed on each thin client workstation.
  1. Installing an IBM Data Server Client or DB2 Connect Personal Edition on the code server.
  2. Making the code directory on the code server available to all thin workstations.
  4. Mapping a network drive from each thin client workstation to the code server.
  5. Running the thnsetup command to setup each thin client.
Response Files Are A Tool To Help Standardize Client Configurations

- Using an exported profile with a response file will automate client installation

Z:\CONPE\setup /U Z:\CONPE\db2conpe.rsp /L C:\db2conpeFP.log /F /T C:\db2conpe_traceFP.log /W
db2iprune

db2iprune is a command line utility that you can use to reduce the size of your DB2 Windows product installation image. This tool is useful for large-scale deployments of DB2, as well as for embedding DB2 within an application.

It consists of an input file and a db2iprune executable. The input file, or ".prn" file, contains a full list of removable components and is used to indicate which features and languages you would like removed from the installation image.

The db2iprune executable (db2iprune.exe) removes the cabinet (.cab) files associated with those features and languages. The result is a new, smaller DB2 installation image that can be installed using the regular DB2 installation methods.

Note: db2iprune is a Windows only program.
db2iprune

- Saved over 200MB on the CD image selecting only needed items…
- Instructions are with the utility…
- Available for selected DB2 LUW Windows V8 FP7+ & V9 products

The file directory were db2iprune is stored:

```bash
?\product name\DB2\WINDOWS\UTILITIES\DB2IPRUNE
```
C:\_pwirth\Software\DB2 V8 Conn PE\CONPE\db2\Windows\utilities\db2iprune> db2iprune -r db2conpe.prn -p "C:\_pwirth\Software\DB2 V8 Conn PE\CONPE" -o "C:\_pwirth\Software\DB2 V8 Conn PE\CONPES_SMALL"

Please wait... The product image is being copied to the destination specified:

C:\_pwirth\Software\DB2 V8 Conn PE\CONPES_SMALL

Deleting .cab files for the feature whose token is SYSTEM_BIND_FILES...
Deleting .cab files for the feature whose token is APPLICATION_DEVELOPMENT_TOOLS...
Deleting .cab files for the feature whose token is LDAP_EXPLOITATION...
Deleting .cab files for the feature whose token is CLIENT_TOOLS...
Deleting .cab files for the feature whose token is DB2_WEB_TOOLS...
Deleting .cab files for the feature whose token is DATA_WAREHOUSE_CENTER...
Deleting .cab files for the feature whose token is INFORMATION_CATALOG_CENTER...
Deleting .cab files for the feature whose token is INFORMATION_CATALOG_CENTER_WEB...
Deleting .cab files for the feature whose token is SPATIAL_EXTENDER_CLIENT_SUPPORT...
Deleting .cab files for the feature whose token is XML_EXTENDER...
Deleting .cab files for the feature whose token is FIRST_STEPS...
Deleting .cab files for the feature whose token is CONFIGURATION_ASSISTANT...
Deleting .cab files for the feature whose token is COMMAND_CENTER...
Deleting .cab files for the feature whose token is CONTROL_CENTER...
Deleting .cab files for the feature whose token is ACTIVITY_MONITOR...
Deleting .cab files for the feature whose token is EVENT_ANALYZER...
Deleting .cab files for the feature whose token is DEVELOPMENT_CENTER...
Deleting .cab files for the feature whose token is DB2_SAMPLE_APPLICATIONS...
Deleting .cab files for the feature whose token is SQLJ_SAMPLES...
Deleting .cab files for the feature whose token is WAREHOUSE_SAMPLES...
Deleting .cab files for the feature whose token is INFORMATION_CATALOG_SAMPLES...
Deleting .cab files for the feature whose token is SPATIAL_EXTENDER_SAMPLES...
Deleting .cab files for the feature whose token is XML_EXTENDER_SAMPLES...
Deleting .cab files for all features of language: BR...
Deleting .cab files for all features of language: CN...
Deleting .cab files for all features of language: CZ...
Deleting .cab files for all features of language: DE...
Deleting .cab files for all features of language: DK...
Deleting .cab files for all features of language: FI...
Deleting .cab files for all features of language: FR...
Deleting .cab files for all features of language: ES...
Deleting .cab files for all features of language: IT...
Deleting .cab files for all features of language: JP...
Deleting .cab files for all features of language: KR...
Deleting .cab files for all features of language: NO...
Deleting .cab files for all features of language: PL...
Deleting .cab files for all features of language: SE...
Deleting .cab files for all features of language: TW...

DOS Command driven…
DB2 Client Tools – Used for DB2 LUW and z/OS

- **Control Center** – is used for administration of DB2/IMS Databases. Control Center can also be used to launch all of the addition DB2 tools.

- **Configuration Assistant** - is used for setting up client/server communications and maintaining registry variables, though it can do more. This is used for DB2 LUW, DB2 400 and DB2 z/OS systems.

- **DB2 Command Line Editor** - is an application you can use to run DB2 commands, operating system commands, or SQL statements.

- **DB2 Command Editor** - Use the DB2 Command Editor to build and execute DB2 commands and SQL statements, and to view a graphical representation of the access plan for an SQL statement.

- **Development Center (V8)** - is used to create business logic for databases (stored procedures and user defined functions). Replaced by IBM Data Studio in V9

- **IBM Data Studio (V9)** - Provides a visual outline of the server database objects grouped by schema. Data Studio can create: database objects, SQL statements and SQL scripts. Provides rich support for XML and Unified debugger for SQL and Java stored procedures.
  
  **Note:** No need for DB2 Connect

- **DB2 for z/OS Optimization Service Center** - Visual Explain functional replacement, expanded explain facility, profile based Query Performance normal/exception monitoring.
  
  **Note:** No need for DB2 Connect

- **DB2® Command Line Processor Plus** (CLPPlus) for dynamically creating, editing, and running SQL statements and scripts. Available in DB2 9.7 IBM Data Server Package.

- The **IBM Database Add-Ins for Microsoft Visual Studio** 2003, 2005 and 2008 are a collection of database functions that integrate into your Visual Studio development environment so that you can work with DB2® servers and develop DB2 procedures, functions, and objects.
Control Center
DB2 Command Line Editor (DB2 LUW 9)
DB2 Command Line Processor (DB2 LUW V8)
Command Window & Command Line Processor (CLP)

Command Window & CLP are handy tools for checking connections to DB2 databases and running scripts. Note that the connection string and commands begin with “db2” in Command Window.
Rapid Application Development Tools...
IBM Data Studio &
IBM .Net Add-Ins for
Microsoft Visual Studio

IBM Data Studio is a handy tool to test Java Type 2 and 4 connections to DB2 databases.
IBM Data Studio

- Eclipse technology
- Integrated Query Editor
- SQLJ Editor
- SQL Builder
- Integrated Query Editor
- SQL Routine Debugger
- Java Routine Debugger
- XML Editor
- XML Schema Editor
- Data Management
- Visual Explain
- Project Management
- Physical Data Diagramming
- Data Distribution Viewer
- Object Management
- Browse & Update Statistics
- Security Access Control
- Connection Management integration with Kerberos and LDAP
- Data Web Services
- IDS Servers Support
- pureQuery for Java*

Notes:
- DB2 z/OS V8 requires APARs PK41262 and PK41138 for support of the SQL-PL debugger
- **IBM Data Studio includes DB2 z/OS Type-4 driver 4, DB2 Connect is not needed**
- Available at: http://www-306.ibm.com/software/data/studio/features.html
DB2 z/OS Optimization Service Center


- Visual Explain functional replacement DB2 V8 and 9 for z/OS
- No-charge offering
- Note: Dec. 2009 fix pack 7 removed the dependency on DB2 Connect
DB2 V8, V9, V9.5 and V9.7 Clients and fix packs:
http://www.ibm.com/support/docview.wss?rs=56&uid=swg27007053
Documentation can be downloaded from:
http://www.ibm.com/support/docview.wss?rs=71&uid=swg27009474
DB2 Client Monitoring

- db2drdat
- DB2 ping
- DB2 Non-Java(cli/ODBC/OLE-DB/ADO)
- DB2 Java
DRDA trace command

- `db2drdat` is shipped with Data Server driver Package from 9.5 FixPack 4 and higher.
- Allows the user to capture the DRDA data stream exchanged between a DRDA Application Requestor (AR) and the DB2 DRDA Application Server (AS). Although this tool is most often used for problem determination, by determining how many sends and receives are required to execute an application, it can also be used for performance tuning in a client/server environment.
DB2 Connect Planning

Check your network connection!

- Simple tools:
  - PING: simple response time measurement. Vary packet sizes (-l parameter)
  - TRACERT: tells you about network hops
  - ROUTE PRINT: see how your TCP/IP routing table
- Check that DDF has sufficient priority
- DB2PING command
  - Like ping but helps you see both network + DDF time
DB2 Client Monitoring: Non-Java

1. Edit the db2cli.ini file used by the CLI/ODBC application, adding the following:
   
   [COMMON]
   Trace=1
   TracePathName=<path>
   TraceComm=1
   TraceFlush=1

2. Execute Program
3. Parse Trace
Sample CLITraceParser Output

- **Overall Trace statistics**
  - 10023 statements in trace.
  - 12.400 seconds total trace time.
  - 0.172 seconds spent for application processing.
  - 12.228 seconds spent for CLI processing.

- **Network specific CLI processing time statistics**
  - 10002 network flows sent to transmit
  - 1230482 bytes, requiring a total of
  - 0.476 seconds.

- 10002 network flows received, transmitting
  - 860386 bytes, requiring a total of
  - 9.214 seconds.

**End of overall trace statistics report**

**Function specific statistics**

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Total</th>
<th>Timing</th>
<th>Application</th>
<th>CLI</th>
<th>Flows</th>
<th>Bytes</th>
<th>Network Send</th>
<th>Time</th>
<th>Network Receive</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQLGetInfo</td>
<td>3</td>
<td>0.001</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>SQLSetConnectAttr</td>
<td>1</td>
<td>0.000</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>SQLExecDirect</td>
<td>1</td>
<td>0.000</td>
<td>0.008</td>
<td>1</td>
<td>10</td>
<td>0</td>
<td>0.000</td>
<td>1</td>
<td>54</td>
<td>0.008</td>
</tr>
<tr>
<td>SQLGetFunctions</td>
<td>1</td>
<td>0.000</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>SQLGetDiagRec</td>
<td>1</td>
<td>0.000</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>SQLPrepare</td>
<td>1</td>
<td>0.000</td>
<td>0.001</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>SQLFreeHandle</td>
<td>3</td>
<td>0.000</td>
<td>0.001</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>SQLAllocHandle</td>
<td>3</td>
<td>0.000</td>
<td>0.002</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>SQLConnect</td>
<td>1</td>
<td>0.000</td>
<td>0.076</td>
<td>1</td>
<td>342</td>
<td>0</td>
<td>0.000</td>
<td>1</td>
<td>208</td>
<td>0.002</td>
</tr>
<tr>
<td>SQLDisconnect</td>
<td>1</td>
<td>0.000</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>SQLExecute</td>
<td>10000</td>
<td>0.170</td>
<td>12.139</td>
<td>10000</td>
<td>1230130</td>
<td>0.476</td>
<td>10000</td>
<td>860124</td>
<td>9.204</td>
<td></td>
</tr>
<tr>
<td>SQLBindParameter</td>
<td>2</td>
<td>0.000</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>SQLGetStmtAttr</td>
<td>4</td>
<td>0.000</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>SQLSetEnvAttr</td>
<td>1</td>
<td>0.000</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
</tbody>
</table>

**End of function specific statistics report**

Network time = (9.214 + 0.476) = 9.69
Application + driver time = 0.172 + 12.228 = 9.69 = 2.71

DB2 Client: Java Monitoring

How to Take a DB2 Universal JDBC Driver Trace

- **Trace as standalone JCC application**
  - *DataSource* interface for connection to JCC
    - `DB2DataSource.setTraceLevel` default `TRACE_ALL`
    - `-javax.sql.DataSource.setLogWriter` `TRACE_ALL` only available
  - *DriverManager* interface for connections to JCC
    - `DriverManager.getConnection`
      - Set the `traceLevel` property in the `info` parameter or `URL` parameter
    - `DriverManager.setLogWriter`
      - Specify trace destination and turn on the trace

- **Within WebSphere, embed the JCC trace points**
  - *Set JDBC trace properties* in WebSphere Application Server
    - Go to Resources > JDBC Provider > Data Sources > Additional Properties > Custom Properties.
    - Set the property: `traceLevel(-1 means full trace TRACE_ALL)`
  - *Turn on the trace*
    - Go to Troubleshooting > Logs and Trace > Pick the server > Diagnostic Trace > Trace Specification: `RRA=all=enabled;WAS.database=all=enabled`
    - Specify two trace strings separated by `:`, one for WAS resource adaptor and one for database (JDBC driver)
Taking JCC Trace Without Modifying the Application (Option 1)

1. Create plain text file on client
   a. named DB2JccConfiguration.properties
   b. with one line of text

2. To enable JCC tracing automatically:
   c. Add file to the CLASSPATH

   ```
   DB2JccConfiguration.properties
   
   db2.jcc.override.traceFile=c:\\jcc.trc
   ```
Taking JCC Trace Without Modifying the Application (Option 2)

1. Create configuration file with trace properties:

```
jcc.properties
  db2.jcc.traceDirectory=c:\\temp
  db2.jcc.traceFile=trace
  db2.jcc.traceFileAppend=false
  db2.jcc.traceLevel=-1
```

2. When Java program is executed, specify the filename via the Option -D

```
java -Ddb2.jcc.propertiesFile=jcc.properties JccTraceExample2
```
   a. The configuration file is placed in same directory as Java class file
      or complete path for configuration file can be specified.
   b. Trace level cannot be specified as constant
JCC Trace Parser *(Available from IBM Field Staff)*

**Features**

- **Trace time analysis**
  - Application processing time
  - JCC processing time
  - Network processing time
  - Time analysis for overall trace
  - Time analysis for overall trace on per connection basis
  - Time analysis per JCC function call

- **Network flow analysis**
  - Bytes sent or received during flush/fill
  - Time taken during flush/fill
  - Total number of flows
  - Network flow analysis on a per JCC function call basis

- **Error report identifying**
  - Prints the entire exception
  - The line number of the error

- **Automatic extraction of multiple traces from a single file**
  - For example, if there are many connections in a single trace file then that many numbers of trace outputs will be generated.
  - To maintain the relationship between connection and respective method calls unique thread name has taken into consideration.

**Input** = Valid JCC trace file  
**Output** = One report file which may contain multiple trace reports depending upon different connections information found in a given trace file
JCCTrace Report
Generated by JCCTraceParser

Overall Trace Statistics for Connection: #1  ThreadName: GR8.thread.pool : 3
===============================================================================
206 methods called in trace.
15031.000 milli seconds total trace time.
2177.000 milli seconds total trace time per connection.
2093.000 milli seconds spent for application processing.
64.000 milli seconds spent for JCC processing.

Network Specific JCC processing time statistics
===============================================================================
16 network flows sent
4256 bytes, and received
4384 bytes, and consumed a total network time of 72.000000 milli seconds.

End of overall trace statistics report

Function specific statistics
===============================================================================

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Total</th>
<th>Application</th>
<th>JCC</th>
<th>Flows</th>
<th>SendBytes</th>
<th>ReceiveBytes</th>
<th>NetworkTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>setMaxRows</td>
<td>14</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>setLong</td>
<td>27</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>close</td>
<td>14</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>getString</td>
<td>38</td>
<td>2.000</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>rollback</td>
<td>4</td>
<td>2086.000</td>
<td>12.000</td>
<td>4</td>
<td>0</td>
<td>128</td>
<td>10.000</td>
</tr>
<tr>
<td>setAutoCommit</td>
<td>2</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>setString</td>
<td>9</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>prepareStatement</td>
<td>1</td>
<td>0.000</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>next</td>
<td>23</td>
<td>2.000</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>setObject</td>
<td>36</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>wasNull</td>
<td>6</td>
<td>0.000</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>executeQuery</td>
<td>14</td>
<td>3.000</td>
<td>72.000</td>
<td>14</td>
<td>4256</td>
<td>4256</td>
<td>62.000</td>
</tr>
<tr>
<td>getTimestamp</td>
<td>2</td>
<td>0.000</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>getLong</td>
<td>18</td>
<td>0.000</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
</tbody>
</table>

End of function specific statistics report
Client Tools - Java Application Monitoring

- Java API for application monitoring
  - `DB2SystemMonitor monitor = ((DB2Connection)conn).getDB2SystemMonitor();`
  - `monitor.enable(true);`
  - `monitor.start(com.ibm.db2.jcc.DB2SystemMonitor.RESET_TIMES);`
  - `monitor.stop();`
  - `monitor.getServerTime();`
  - `monitor.getNetworkIOTime();`
  - `monitor.getCoreDriverTime();`
  - `monitor.getApplicationTime();`

See
z/OS manual:
SC18-7414
Redbook:
SG24-6435
Agenda

- Topology of Distributed Applications
- DB2 Clients
- DB2 Connect
- DB2 z/OS and Distributed Connections
- DB2 z/OS and Connection Resources
- z/OS Workload Manager
- DB2 Private Protocol
- Supplemental Materials
What are we talking about on the “Roadmap”?

1. Web Application server with DB2 Connect or Type-4 Java Driver or IBM Data Server Driver

2. DB2 Connect Gateway server on zLinux and Hypersockets

3. Virtual IP Addressing (VIPA) & SysPlex Distributor

Excel

IBM Client

Internet Explorer

Client

Web Application server with DB2 Connect or Type-4 Java Driver or IBM Data Server Driver

IP Sprayer (Load Balancing)

Edge server

Client

Web Application server

IBM Client

Virtual IP Addressing (VIPA) & SysPlex Distributor

DB2 Connect Gateway server on zLinux and Hypersockets
DB2 Connect

- DB2 Connect provides Distributed Relational Database Architecture (DRDA) Application Requester (AR) function. With Communications Support, DB2 Connect enables applications running on DB2 UDB for UNIX, OS/2, and Windows client platforms to access and update data on the following:
  - DB2 for MVS
  - DB2 for OS/390
  - DB2 for VM and VSE
  - DB2 for OS/400
  - DRDA compliant database *management systems*
  - Two version: DB2 Connect Personal Edition and DB2 Connect Server Edition available
Why Use DB2 Connect

- Provides high availability:
  - Data Sharing Support
  - Client reroute –
    
    ```
    db2 update alternate server for database mydb using hostname
    mydb2.myzos.com port 456
    ```

- Connection Control
- Connection Concentration
- Monitoring by DB2PE agent
- Multi-transport models such as Tuxedo, WebLogic, and Encina still require a middle-tier DB2 Connect Server
DB2 Connect
In a nutshell

- Desktop PCs
- Application servers
- Web application servers

APIs
Communication
Data

- DB2 Connect functions
DB2 Connect
3 tiers of a solution
DB2 Connect Server
Basic Architecture Overview

- `db2start` starts instance
- Multiple instances/machine
- 1-1-1 relationship between applications, coordinating agents and DB2 threads
- Capacity of the server is determined by the MAXAGENTS (1 to 64K) (default 200)
- `LIST DCS APPLICATIONS` display info for all agents
- Don’t set MAXAGENTS so high that it would cause your connect server to page
- SQL1226N - Issued by DB2 Connect if the maximum number of agents has already been started

DB2 Connect Agents:
- On Unix, each agent is a process consuming approximately 700 K of memory.
- On Windows, each agent is a thread consuming approximately 300 K of memory.
DB2 Connect Server
Connection Pooling Overview

- 1-1-1 relationship between applications, coordinating agents and DB2 threads
- Initial pool size is controlled by NUM_INITAGENTS
- Max pool size is controlled by NUM_POOLAGENTS
- Coordinating Agents (CA) and corresponding connection returned to the pool on disconnect
- NUM_POOLAGENTS is a performance parameter; this parameter is used to avoid agent creation for every new connection

C:> db2start
C:>
DB2 Connect Server
Connection Concentrator

- N-1 relationship between applications and threads
- CA and corresponding connection returned to the pool on commit/rollback
- Connection concentrator is activated when MAX_CONNECTIONS > MAX_COORDAGENT
- Initial pool size is controlled by NUM_INITAGENT
- CA=Coordinating Agents
- CO=Connections

Note: The MAX_LOGICAGENT parameter was renamed to MAX_CONNECTIONS in V8

The precise ratio of MAX_CONNECTIONS to MAX_COORDAGENT that you can achieve depends on the workload characteristics. As a rule of thumb, you can start with a 5:1 ratio, and tune from that point, using the monitoring techniques described in 6.2, “Thread monitoring” on page 163 SG24-6952-00.
DB2 Connect Connection Concentrator
Reduce mainframe resource usage (Saves on DBATs)

- Connection Concentrator reduces number of DB2 threads
- Saving a thread saves:
  - Active: 250K
  - Inactive: 7K
- You decide how many threads to keep around by setting DB2 Connect parameter MAXCOORDAGENTS
- There is a cost (special registers and user id switching)
Sysplex + Concentrator
24*7 for the DB2 for OS/390

- DB2 datasharing: DB2 cluster for high availability
- DB2 Connect Connection
  Concentrator is datasharing aware:
  - Balances workload
  - Routes transactions around failed members

To enable DB2 SYSPLEX (data sharing) support; set the sixth parameter to ‘SYSPLEX’ in the DCS catalog parameter string for the cataloged database on the DB2 Connect Server. To
So what are the key points to Pooling and Concentrator

- Booth can support Data Sharing
- Booth can support VIPA and Sysplex Distributor
- Connection Pooling reuses threads at connection time
- Connection Concentrator reuses threads at commit time

  - Restrictions:
    - Does not support SSL
    - If you declare global temporary tables, they must be closed explicitly at transaction or branch boundary
    - Only supports dynamic SQL from the Call Level Interface (CLI)
    - Dynamic prepare requests from embedded dynamic SQL applications will be rejected.
    - See DB2 Connect user guide for details on other restrictions
    - SAP does not support Connection Concentrator

  - Personal Observations:
    - Most customers use Connection Pooling
    - Thoroughly test usage of Connection Concentrator – often times applications are using some of the above restrictions which do not show up until in production
    - Be current on DB2 Connect fix pack, many improvements to concentrator code over time
How do I monitor agent activity?
Two tools DB2 snapshot monitor or db2pd

Example of db2pd agent information:

SYSPLEX host information is available using: db2pd –sysplex
Can I monitor connection activity?

Step 1 – Setup monitoring
Can I monitor connection activity? (continued…)

Step 2 – Get snapshot…

```
C:\Program Files\IBM\SQLLIB\BIN>db2 get snapshot for dcs applications on ndc2db20 3

DCS Application Snapshot

Client application ID = 192.168.15.107.6916.07053003343
Sequence number = 00001
Authorization ID = DBA031
Trusted Authorization ID =
Application name = devenv.exe
Application handle = 31
Application status = waiting for request

Status Client
Client

Client

Info for statements taking 2 transmissions:
Number of SQL statements = 9
Client Total Outbound bytes sent = 2246
Process Total Outbound bytes received = 9312
Client Outbound bytes sent high water mark = 1441
Host application Outbound bytes received high water mark = 4128
Sequence Outbound bytes sent low water mark = 10
Database Outbound bytes received low water mark = 54
DCS data
Host data
Host response
Total Host response time (sec.ms) = 0.627695
Total elapsed times for stmts (sec.ms) = 0.004301

Number of statements with outbound bytes sent:
Between 1 and 128 bytes = 3
Between 129 and 256 bytes = 2
Between 257 and 512 bytes = 1
Between 513 and 1024 bytes = 0
First byte
Outbound bytes
Outbound bytes
Inbound bytes
First byte
Time spent
Last response
Last response
Last response
Number of bytes

```
DB2 Universal Driver For Java - Type 4 Driver

- IBM DB2 Driver for JDBC and SQLJ provides connection pool and connection concentrator functionality similar to DB2 Connect (DB2 V8 FP10 and later)
- Ability to reuse server agents at application commit or rollback
Which Connection Pool Should I use?

- To exploit DB2 Data Sharing workload balancing and transparent failover, both, application server connection pool AND connection concentrator/ connection pool should be used
**DB2 Universal Driver Connection Concentrator**

- Global properties defined in Global Properties File:
  - `db2.jcc.maxTransportObjects` - equivalent to MAXAGENTS - max # of connections to DB2 server across all datasources (default value is -1, meaning no limit)
  - `db2.jcc.minTransportObjects` - equivalent to NUM_POOLAGENTS - # of connections kept in the pool across all datasources - # of transport objects will grow as requested but always stay (default value is 0)
  - `db2.jcc.maxTransportObjectIdleTime` - time in sec., a connection stays idle in the pool before it is closed, until minTransportObject is reached (default value is 60 sec)
  - `db2.jcc.maxTransportObjectWaitTime` - if maxTransportObjects is reached - time in sec., an application waits to get a connection before throwing a SQLException (default value is 5 sec)

- DataSource Properties:
  - `maxTransportObjects` - max # of connections to DB2 server from this DataSource. Can not be bigger than `db2.jcc.maxTransportObjects` (default value is -1, meaning no limit)
  - `enableConnectionConcentrator`** - enables connection concentrator functionality. Not allowed for DB2 LUW (default value is "false" - disabled)
  - `enableSysplexWLB` - enables Sysplex Workload Balancing functionality. Not allowed for DB2 LUW (default value is "false" - disabled)

** Helps saves DBATs on DB2 z/OS
WebSphere Connection Pooling . . .
The Java Type 4 Driver

- **Make sure Java DB pool and JDBC Driver Logical Connects agree**
- Connection Concentrator is independent of Sysplex Workload Balancer, either feature can be set on or both
- **Connection Concentrator**
  - Heavyweight Reuse vs Lightweight Reuse
  - Heavyweight Reuse avoided when there are multiple commit scopes within a connect scope. If this isn’t the case, do not use connection concentration, just WebSphere connection pooling
  - Use JCC global transport pool statistics
- **Sysplex Workload Balancer**
  - Unlike the DB2 Connect gateway, SWB does not depend on connection concentration being activated
  - Use VIPA to always assure a 1st connection
  - WLM sends back a server list
  - Workload balancing as well as failover
  - DB2 data sharing group name must be specified as the WebSphere datasource database
- **Subsetting Alias can be referenced by application instead of the BSDS location name**
- Both features are turned off by default
How can I monitor Type 4 driver activity

- To assist you in monitoring the performance of your applications with the DB2 Universal JDBC Driver, the DB2SystemMonitor interface is provided. This interface contains methods that collect the data about a connection.
- To monitor the DB2 Universal JDBC Driver connection concentrator and Sysplex workload balancing, you need to monitor the global transport objects pool. You can monitor the global transport objects pool in either of the following ways:
  - Using traces that you start by setting DB2 Universal JDBC Driver configuration properties
  - Using an application programming interface
  - Configuration properties for monitoring the global transport objects pool:

See DB2 Manual: Application Programming Guide and Reference For Java
Sysplex Workload Balancing using the IBM Data Server Driver

Note: CLP does not support Sysplex WLB
Properties for fine-tuning Sysplex workload balancing for direct connections from non-Java applications to DB2 for z/OS

<table>
<thead>
<tr>
<th>Element in the db2dsdriver configuration file</th>
<th>Section in the db2dsdriver file</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxTransportIdleTime</td>
<td>WLB</td>
<td>Specifies the maximum elapsed time in number of seconds before an idle transport is dropped. The default is 600. The minimum supported value is 0.</td>
</tr>
<tr>
<td>maxTransportWaitTime</td>
<td>WLB</td>
<td>Specifies the number of seconds that the client waits for a transport to become available. The default is -1 (unlimited). The minimum supported value is 0.</td>
</tr>
<tr>
<td>maxTransports</td>
<td>WLB</td>
<td>Specifies the maximum number of connections that the requester can make to the data sharing group.</td>
</tr>
<tr>
<td>maxRefreshInterval</td>
<td>WLB</td>
<td>Specifies the maximum elapsed time in number of seconds before the server list is refreshed. The default is 30. The minimum supported value is 0.</td>
</tr>
</tbody>
</table>
Properties for enabling only Sysplex workload balancing for connections from non-Java applications to DB2 for z/OS

<table>
<thead>
<tr>
<th>Element in the db2dsdriver configuration file</th>
<th>Section in the db2dsdriver file</th>
<th>Description</th>
<th>Value to set</th>
</tr>
</thead>
<tbody>
<tr>
<td>enableWLB</td>
<td>WLB</td>
<td>Specifies whether workload balancing is enabled.</td>
<td>true</td>
</tr>
<tr>
<td>enableAcr</td>
<td>ACR</td>
<td>Specifies whether automatic client reroute is enabled. For CLI or .NET applications, enabling automatic client reroute automatically enables seamless failover.</td>
<td>false</td>
</tr>
<tr>
<td>enableSeamlessAcr</td>
<td>ACR</td>
<td>Specifies whether seamless failover is enabled. Among non-Java applications, seamless failover is supported only for CLI or .NET applications. true is the default.</td>
<td>If enableAcr is false, this value is false, so you do not need to set it.</td>
</tr>
</tbody>
</table>
Sample WLB Configuration
(part of db2dsdriver.cfg)

<databases>
  <database name="STLEC1" host="9.30.30.5" port="446">
    <WLB>
      <parameter name="enableWLB" value="true"/>
      <parameter name="maxTransports" value="100"/>
      <parameter name="maxTransportIdleTime" value="600"/>
      <parameter name="maxTransportWaitTime" value="30"/>
      <parameter name="maxRefreshInterval" value="30"/>
    </WLB>
    <ACR>
      <parameter name="enableACR" value="true"/>
      <parameter name="enableSeamlessACR" value="true"/>
    </ACR>
  </database>
</databases>
DRDA trace command

- `db2drdat` is shipped with Data Server driver Package from 9.5 FixPack 4 and higher.
- Allows the user to capture the DRDA data stream exchanged between a DRDA Application Requestor (AR) and the DB2 DRDA Application Server (AS). Although this tool is most often used for problem determination, by determining how many sends and receives are required to execute an application, it can also be used for performance tuning in a client/server environment.
Driver vs DB2 Connect Gateways

- If a Driver connects through a DB2 Connect gateway, then JCC connection concentration and WSB are ‘turned off’ at the JCC level
  - **Driver**
    - Better granularity of transport agents that each JVM can use
    - Each Driver can specify the maximum number of allowed transport objects (which = DBATs) per datasource
    - Slightly better performance (1-2% faster)
    - Eliminates the need for a middle-tier DB2 Connect Server.
    - New WLB algorithm has equivalent features to the DB2 Connect Connection Concentrator (enabled by default).
  - **Gateway**
    - Bigger Scope – connection concentration supports multiple JVMs from multiple WAS servers … bigger DBAT savings
    - Easier to manage upgrades and changes in one place (one gateway versus multiple JCC drivers)
    - Multi-transport models such as Tuxedo, WebLogic, and Encina still require a middle-tier DB2 Connect Server
    - Software entitlement costs
    - Provides additional monitoring – But DB2 10 for z/OS will have improvements
Security Considerations

- Industrial strength authentication techniques
  - SERVER_ENCRYPT_AES (Recommended Setting)
    - Default
    - New in 9.5 FixPack 3
    - Strong 256-bit AES encryption (need DB2 for z/OS APAR PK56287)
    - No noticeable performance penalty
  - KERBEROS (third-party authentication)
    - Set TCPALVER to NO (DB2 installation parameter)

- Network Encryption
  - SSL (Connection-based security using HTTPS protocol)
    - DB2 9 for z/OS uses the z/OS Communication Server (z/OS CS) IP Application Transparent Transport Layer Security (AT-TLS).
    - Configuration and setup required at both server and client.
    - To enable SSL for Java connections, use properties.put("sslConnection", "true")
    - When using the db2dsdriver.cfg add <parameter name="SecurityTransportMode" value="SSL"
  - IPSec (Host-based security)
    - An open architecture for security at the IP networking-layer.
    - No application modifications.
    - Eligible for zIIP processing.
  - DATA_ENCRYPT (driver setting)
    - Not considered secure and Not recommended.
IDLE THREAD TIMEOUT – ZPARAM=IDTHTOIN

- Code 00D3003B is issued when the IDTHTOIN time limit is exceeded and a thread is canceled. The thread is canceled after the timeout value expires; its locks and cursors are released. Inactive and in doubt threads are not subject to time-out.
- If a DB2 Client tries using a canceled thread it will get either of the following messages:

Before: Version 8.2 FixPak 4 (equivalent to Version 8.1 FixPak 11)
- [IBM][CLI Driver][DB2] SQL30081N Communication function detecting the error: ”recv”…. Protocol specific error code(s): "+", "+", "0". SQLSTATE=08001
  If <protocol> = TCP/IP, <rc1>=*, <rc2>=*, and <rc3>=0, Then the remote database has terminated the connection

After: Version 8.2 FixPak 4 (equivalent to Version 8.1 FixPak 11)
- [IBM][CLI Driver][DB2] SQL30108N A connection failed but has been re-established…. Special registers may or may not be re-attempted (Reason code = "2"). SQLSTATE=08506
- DB2_MAX_CLIENT_CONNRETRIES: The maximum number of connection retries attempted by automatic client reroute was added and will automatically retry the connection, so there is a little change in the message.

- Tip: To try and avoid the DB2 Client messages SQL30081N & SQL30108N, see if the client application has a “Client” timeout parameter. If there is: then set the client timeout lower than the DB2 z/OS IDTHTOIN parameter. Be careful changing this parameter - because it is for all DDF applications. It was 0 in V7 and is now 120 seconds in V8.
Client Query Timeout

- SQL Interrupt (soft CANCEL) feature introduced in V8 may be incompatible with DDF applications using CLI / ODBC.
  - An application timeout, which would cause the termination of the thread prior to V8, now causes the termination interruption of the SQL statement, which is a behavior change that may introduce application issue.
  - The default value of Microsoft CommandTimeout is 30 seconds
  - ODBC: SQL_ATTR_QUERY_TIMEOUT
  - ADO: CommandTimeout
- Possible work around options are to increase timeout thresholds and / or apply PK41661-V8 or PK59385-V9 to disable this feature via a new ZParm, until the applications can be adjusted.
  - ZParm: SQLINTRP - SQL Interrupt processing ENABLEd or DISABLEd
- DB2 client QueryTimeoutInterval can override by setting to zero also
- When client time outs happen you may see “soft cancels” in the DB2 z/OS log: 00E50013. No dump is issued and the soft cancel allows DB2 to backout or cleanup resources. Some times these cancels come in groups because resources were unavailable for a long time; like a batch program that does not commit updates very often.
Additional DB2 Client and Connect Settings

Client Settings:

- **CLISCHEMA=XXX** A shadow catalog can be created by using the DB2OCAT tool from IBM at the tools site (ftp://ftp.software.ibm.com/ps/products/db2/tools/) . This shadow catalog can be maintained automatically by DPROP. This speeds up catalog processing for ODBC and JDBC applications.
- **AutoCommit Usage.** By default, autocommit=1 meaning a commit will flow automatically in the same block in which the SQL is sent to the host. This is good for OLTP environments. But this can be very bad for applications that do a huge amount of INSERTs.
- **Schemalist=’Payroll’, ‘Accounting’** SQL submitted through a dbname where a schema list is specified will be further qualified to only access the tables with the schema’s specified. This can speed up system catalog access considerably.
- **Dbname=PRODUCTION.** Same idea as schemalist, except with regard to a database.
- **DB2_ENABLE_LDAP=Y** within the DB2 Registry can elongate connect times for unpooled connections, so set it to N if not needed. Many circumstances can cause the need for an unpooled connection to be created. These circumstances are detailed in the [DB2 Connect User’s Guide](#).

**NOTE:** *Avoid* editing db2cli.ini manually, parameters do have requirements of position in the file, use DB2 Configuration Assistant to avoid mistakes
Additional DB2 Client and Connect Settings

Client Settings (Continued…):

- Explicitly bind the DDCSMVS.LST with BLOCKING ALL with a userid that has bindadd authority in the NULLID collection. If you don’t do this and someone connects to the target DB2 for zOS subsystem who just happens to have this needed authority and this person runs a query then the DDCSMVS.LST packages will be bound implicitly without BLOCKING. All subsequent users of the gateway will experience poor query performance since rows will be returned on a one row per block basis.
- Always use DB2BATCH, not the CLP, when estimating performance from the client desktop to the host DB2 subsystem.
- MS ACCESS does not use the most efficient ODBC instruction for pulling data from a database. Other tools vendors (even MS Query) have corrected this problem a long time ago. Consequently, for large result sets that query will appear to run much faster in any query tool than in MS ACCESS.
- Microsoft applications often have SQL_ATTR_QUERY_TIMEOUT set to 30, and therefore terminate before any DB2 z/OS timeout parameters (SQLSTATE 57014 - Processing was canceled as requested). DB2 client configuration keyword “querytimeoutinterval” can tune off this timeout. The QueryTimeoutInterval configuration keyword is used to indicate how long the CLI driver should wait between checks to see if the query has completed, this parameter can be set to “0” to disable client time outs.
Additional DB2 Client and Connect Settings…

DB2 Connect Settings:

- TCPKEEPALIVE setting in operating system on which the DB2 Connect EE software runs is important in eliminating DB2 hanging threads under certain circumstances. If tcpipkeepalive requests by that operating system are not responded to, then DB2 Connect running on that operating system will tell DB2 on zOS to kill the DB2 threads that were being used by the applications on the non-responding platforms. A five minute setting is suggested.
- Configure enough memory on the DB2 Connect gateway to prevent paging.
- When connecting DB2 Connect EE on the server as MTS or WAS, set the following DB2 registry variable `db2connect_in_app_process=n`. This will allow local applications to connect through the DB2 instance. You must do a db2start in order to accomplish connectivity. This will also allow local connections to take advantage of two other performance enhancements within DB2 Connect, namely connection pooling and connection concentration. Since local connections will appear to DB2 Connect as though they were remote connections, you can use the LIST DCS APPLICATIONS and GET SNAPSHOT FOR ALL DCA APPLICATIONS.
- TCPIPRECEIVEWINDOW (NT – 8k, UNIX – 16k defaults) – raise to 64k or better invoke tcpip window scaling by issuing a: `db2set –g db2sosndbuf=65537 and db2set –g db2sorcvbuf=65537` if your operating system supports it. Check the db2diag.log as this may be done automatically as of DB2 Connect V8.1. This raises to 64k, the amount of data that can be transmitted between the mainframe and DB2 Connect gateway as well as gateway and client, before an acknowledgment must be sent.
- Remove the queuing that DELAYACK (on DB2 for zOS IP port as well on the DB2 Connect IP port), that affects incoming traffic.
Additional DB2 Client and Connect Settings…

IP Network Settings:

- CISCO Router defaults, eg. + PROTOCOL BLOCKING=yes. Change to no!!! If no, then traffic is queued for up to 5 ms.
- Firewall port timeout values for firewalls between the mainframe and the DB2 Connect gateway … set to never timeout. or you'll see DSNL511I error messages on the mainframe DB2 log.
- Change the default incoming Queue Length on the device card in DB2 Connect gateways UNIX On UNIX systems (especially AIX), this is accomplished by changing the device characteristics such as Transmit and/or Receive queue depth. The defaults are 32 and a recommended value is 150.
- As mentioned before, change the TCPIPKEEPALIVE in DDF and on DB2 Connect gateway.

DB2 9 for z/OS Features:

- Use Progressive Streaming (also called Dynamic Data Format Data) for LOBs and XML data.
  - Default when using Data Server Drivers.
  - Small LOBs are returned along with non-LOB data.
- Use multi-row fetch (rowset cursors)
  - Java driver uses MRF by default for scrollable cursors ((setFetchSize() can be used to limit number of rows).
  - CLI driver uses DB2BulkOperations.
- Use multi-row insert (atomic/non-atomic)
  - Java batch updates (addBatch())
  - CLI Array Input Chaining (SQL_ATTR_CHAINING_BEGIN/END)
  - .NET DB2BulkCopy
Additional DB2 Client and Connect Settings…

Application Coding Practices:

- Use parameter markers instead of literals with SQL (set Patch2=18 for automatic insert of SQL literal to parameter marker substitution. For SELECT SQL, programmers must code parameter markers. This makes Access Plan reuse made possible on DB2 for zOS.
- Use the SET CLIENT INFO api or java method within applications. This is useful for identifying pooled threads, doing performance or problem diagnosis. It is also possible to write a WLM policy to give better performance for certain users or applications.
- Only bring back the rows needed within the application by limiting results with predicates. Sort and aggregate using SQL, not your program. Use SQL and api constructs to limit rows by using: Fetch First n Rows, Optimize for n Rows, ODBC: set SQL_MAX_ROWS statement option, or JDBC: setMaxRows on a statement object.
- Use embedded SQL instead of dynamic SQL to eliminate access plan overhead.
- Avoid the use of CURSOR WITH HOLD unless needed.
- **Commit your work often, commit even if you are reading**, commit on return from stored procedures, exception: mass inserts then use appropriate commit scope, control your commit scope within your program instead of relying on autocommit.
- For bulk insert, update, delete activity use COMPOUND SQL to reduce network time, but remember DB2 for zOS supports non-atomic COMPOUND SQL so your application is responsible for transaction scope.
- Use CLI trace for problem determination and performance tuning. This is supported for CLI, ODBC, OLEDB and .NET Data Provider environments. Use JDBC tracing for java environments.
Additional DB2 Client and Connect Settings…

Application Coding Practices (Continued):

- Use remote stored procedures.
  - Native SQL procedures called via distributed clients are zIIP-eligible.
  - Use result set cursors to return data.
  - Use COMMIT on RETURN clause for stored procedures that are not nested and do not return result sets.
- Free resources you no longer need – close connections, explicitly close cursors declared WITH HOLD, declare DGTTs with ON COMMIT DROP TABLE, free LOB locators etc.
- Use KEEPDYNAMIC(YES) for applications that use very few SQL statements very often to avoid excessive prepares and keep in mind that it prevents the connection from being inactivated and threads being pooled. (potential for storage build up)
- Consider using static SQL (pureQuery/SQLJ) to get performance and security benefits over dynamic SQL.
- Help the server use limited block fetch (and extra blocks) by using OPTIMIZE for n ROWS and FETCH FIRST n ROWS ONLY.
  - SQL_ATTR_MAX_ROWS for CLI apps.
Additional Connection Information DB2 z/OS

- A DB2 Universal Database for z/OS and OS/390 subsystem identified by its LOCATION NAME or one of the alias LOCATION names defined on the z/OS server. The LOCATION NAME can be determined by the DSNL004I message (LOCATION=location), which is written when the Distributed Data Facility (DDF) is started. The -DISplay DDF command could also be used.

- If accessing a z/OS data sharing group, the domain name should map to the DB2 group dynamic VIPA address. This address routes to the least loaded DB2 member. To access a specific member use the specific DB2 member dynamic VIPA address and turn off sysplex routing. Each member DSNL004I message displays the member specific domain name.

- The 6th positional parameter on the DCS entry, can be used to explicitly enable DB2 Connect SYSPLEX support for a particular database.
Agenda

- Topology of Distributed Applications
- DB2 Clients
- DB2 Connect
- DB2 z/OS and Distributed Connections
- DB2 z/OS and Connection Resources
- z/OS Workload Manager
- DB2 Private Protocol
- Supplemental Materials
What are we talking about on the “Roadmap”? 

1. Web Application server
   - With DB2 Connect or Type-4 Java Driver or IBM Data Server Driver
   - IP Sprayer (Load Balancing)

2. DB2 Connect Gateway server

3. Virtual IP Addressing (VIPA) & Sysplex Distributor
   - zIIP
   - DB2 Connect Gateway server on zLinux and Hypersockets
   - WLM
   - Virtual IP Addressing (VIPA) & Sysplex Distributor

Client
- Internet Explorer
- IBM Client
- Excel

Web Application server
- IBM Client

Edge server

Client
- IBM Client
- Web Application server
DB2 Configuration:

The next two sections will review workload impact on DB2, how to limit workload, and monitor the amount resources used and that are need by DB2 to support distributed transactions. These sections basically explain how to defend DB2 for availability and resiliency.
DB2 z/OS Thread Pooling – In addition to DB2 Connect Pooling

DB2 thread pooling is a mechanism to share DBATs between connected applications. It is implemented wholly within DB2 for z/OS and OS/390, and can be exploited by connections from any DRDA Application Requestor. This support is sometimes called type 2 inactive thread support, but DB2 thread pooling is a far better and more accurate description of this type of pooling.

The benefits for DB2 z/OS Thread Pooling are:
- CPU savings in DB2, by avoiding repeated creation and destruction of DBATs
- Real memory savings in z/OS, by reducing the number of DBATs
- Virtual storage savings in DBM1, by reducing the number of DBATs
- Greater capacity to support DRDA connections
DB2 z/OS V8 Address Spaces: Zooming in on DBM1 & DDF

Threads:
- DBATs
- CONDBATs
# Key DB2 DDF Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Possible values</th>
<th>Default V8 &amp; V9</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDF</td>
<td>NO / AUTO / COMMAND</td>
<td>No</td>
<td>DDF Startup</td>
</tr>
<tr>
<td>CMTSTAT</td>
<td>ACTIVE / INACTIVE</td>
<td>INACTIVE</td>
<td>Thread Pooling - Pool inactive threads</td>
</tr>
<tr>
<td>CTHREAD</td>
<td>1-2,000</td>
<td>200</td>
<td>Max users - allied (local) threads RRSAF or CAF (CICS, IMS, TSO, Batch attach, SPUFI, Classic QMF, etc)</td>
</tr>
<tr>
<td>MAXDBAT</td>
<td>0-1,999</td>
<td>200</td>
<td>Max remote active DDF Threads - DBM1 Address Space</td>
</tr>
<tr>
<td>CONDBAT</td>
<td>0-15,000</td>
<td>10,000</td>
<td>Max remote connections - DDF Address Space</td>
</tr>
<tr>
<td>MAXTYPE1</td>
<td>0-CONDBAT</td>
<td>0</td>
<td>Max inactive DBATs, these are used for private protocol. DRDA uses inactive connections.</td>
</tr>
<tr>
<td>POOLINAC</td>
<td>0-9,999</td>
<td>120</td>
<td>Approximate time, in seconds that an inactive/unused DBAT can remain idle in the pool before it is terminated. DBAT deleted after being used 200 times also.</td>
</tr>
<tr>
<td>IDTHTOIN</td>
<td>0-9,999</td>
<td>120</td>
<td>The IDLE THREAD TIMEOUT (IDTHTOIN) parameter specifies the time (in seconds) that an active DBAT can remain idle before it is canceled. – Should set a “little” higher than TCPKPALV</td>
</tr>
<tr>
<td>TCPKPALV</td>
<td>ENABLE / DISABLE / 1-65534</td>
<td>120</td>
<td>TCP/IP keep alive (Goes hand-in-hand with IDTHTOIN)</td>
</tr>
<tr>
<td>CONTSTOR</td>
<td>YES / NO</td>
<td>No</td>
<td>Periodically &quot;contract” each thread’s working storage area.</td>
</tr>
<tr>
<td>MINSTOR</td>
<td>YES / NO</td>
<td>V8 = No V9 = Yes</td>
<td>Use storage management algorithms that minimize the amount of working storage consumed by individual threads.</td>
</tr>
</tbody>
</table>
Thread Types:

- There are two kinds of inactive threads used in DB2 for OS/390 and z/OS, called type 1 (Private Protocol) and type 2 inactive threads. Type 2 inactive threads are only for DRDA connections, and type 2 inactive threads are preferred over type 1 inactive threads.

- A type 1 inactive thread is really an inactive thread. If a thread becomes type 1 inactive, the DBAT remains in DBM1, and remains linked to the connection information in the DDF address space. The only thing that happens when the thread becomes type 1 inactive, is that DB2 tries to clean up as much storage that is no longer in used or required by the thread as possible.

- Type 2 inactive threads, on the other hand, are not really inactive threads. They are inactive connections in the DDF address space. At commit time the connection information inside DDF is separated from the DBAT. The DBAT itself is returned to a pool of (inactive) DBATs and can be reused by another connection.

DB2 z/OS V8 renamed Type 1 / 2 Inactive Threads to: Inactive DBATs & Inactive Connections (respectively)
Thread Monitoring:

- **SQL1226N** - Issued by DB2 Connect if the maximum number of agents has already been started

- **-DIS DDF DETAIL** provides real time information on DBAT and CONDBAT metrics

- **Hung connection** MAXDBAT is reached, a new connection request to DB2 for z/OS and OS/390 will simply wait for a DBAT to become available...forever. Using the DB2 Statistics Trace and tools like DB2 PE will report the number of times that DBATs queued. No error message issued. Customers complain system slow...

- **Maximum concurrent connections** - represents the number of times the maximum number of connections (CONDBAT) was reached. DB2 PE will report this condition also. Customers complain system slow...
APAR-PK01230: DISPLAY THREAD(*) SERVICE(WAIT) BOOST ENHANCEMENT

- The “-DISPLAY THREAD(*) SERVICE(WAIT)” command can issue a priority boost for agents holding a latch for 2x IRLM timeout limit or a minimum of 1 minute.
- An MVS SYSEVENT ENQHOLD enhancement was needed for DB2 to boost agents effectively.
- Problem conclusion:
  - The -DISPLAY THREAD(*) SERVICE(WAIT) command exploits the SYSEVENT ENQHOLD enhancements introduced in APAR OA08949.
  - By using this boost DB2 will:
    1. Not deboost an agent after issuing a priority boost. The boost is now self-expiring.
    2. Boost at enclave and address space granularity.
    3. Not be susceptible to timing windows.
Deadlocks: ZPARAM=IRLMRWT

**IRLMRWT** Specify the number of seconds before a time-out is detected. *Time-out* means that a lock request has waited for a resource (or for claims on a resource for a particular claim class to be released) longer than the number of seconds specified on this option. The value that is specified for this option must be a multiple of the DEADLOCK TIME on installation panel DSNTIPJ because IRLM uses its deadlock timer to initiate time-out detection and deadlock detection. This value is rarely the actual time.

**IRLM DEADLOCK TIME** Specify the time, in seconds or milliseconds, of the local deadlock detection cycle.

DB2 z/OS reason code "00C9008E" is issued when **IRLMRWT** time is exceed. The following messages are issued in:

**DB2 LUW:**
SQL0913N Unsuccessful execution caused by deadlock or timeout. Reason code "00C9008E". SQLSTATE=57033

**DB2 z/OS:**
DSNT376I ! PLAN=DISTSERV WITH 547
  CORRELATION-ID=db2bp.exe
  CONNECTION-ID=SERVER
  LUW-ID=G943B286.HE0D.070918144329=31047
  THREAD-INFO=DBA031:IBM-WIRTHP:dba031:db2bp.exe
  IS TIMED OUT. ONE HOLDER OF THE RESOURCE IS PLAN=DISTSERV WITH
  CORRELATION-ID=db2bp.exe
  CONNECTION-ID=SERVER
  LUW-ID=G943B286.D610.070918142843=31036
  THREAD-INFO=DBA031:IBM-WIRTHP:dba031:db2bp.exe
  ON MEMBER DSNC
DSNT501I ! DSNIILMCL RESOURCE UNAVAILABLE 548
  CORRELATION-ID=db2bp.exe
  CONNECTION-ID=SERVER
  LUW-ID=G943B286.HE0D.070918144329=449072
  REASON 00C9008E
  TYPE 00000302
  NAME DSNDB04.EMPL177J.X'000002'
TCPIP Keepalive Considerations

- Set TCPKPALV in DDF parms
  - DB2 can detect when DB2 Connect gateways die
  - DB2 can clean up affiliated DBATs
  - z/OS default is several hours
- Set TCPKEEPALIVE on DB2 Connect server’s operating system
  - DB2 Connect server can detect when desktops and Web Application servers die
  - DB2 can clean up affiliated DBATs
- More information on side KEEPALIVE settings are at:
  http://www.ibm.com/support/docview.wss?fdoc=imdb2luw&rs=71&uid=swg21231084
How do I figure out thread activity?
Some type of monitor tool is needed like Omegamon or DB2 PE (like below) is needed.

```
CONTACT          VTYM   Q2      V410./C DB1S 09/22/08 17:04:53 2
> Help PF1 Back PF3 Up PF7 Down PF8 Sort PF10 Zoom PF11
> T.A          OMEGA VIEW PA2
> THREAD ACTIVITY: Enter a selection letter on the top line.
>
> *-ALL B-TSO C-CICS D-IMS E-BACKGROUND F-DIST ALLIED
> G-DIST DBAC H-UTIL I-INACT J-FILTER K-FUNCTIONS L-STORED PROC
> M-TRIGGERS N-SYSPLEX O-ENCLAVES P-WORKSTA

---------------------------------------------------------------------------------------
> ALL THREADS CONNECTED TO DB2

FLTR ON
+ *
+ Elapsed Package   CPU Status GetPg Update Commit CORRID
+ ----------- ------ ---- --------- ------ ------------ -----------
+ 00:02:46.5 SYSLN200 00.0% IN-DB2 1805   0 0 db2jcc_appli
+ 00:02:32.4 SYSLN200 00.0% IN-DB2  1615   0 0 db2jcc_appli
+ 00:02:23.1 SYSLN200 00.0% IN-DB2  1572   0 0 db2jcc_appli
+ 00:00:40.9 SYSLN200 00.0% IN-DB2   286   0 0 db2jcc_appli
+ 00:00:21.3 SYSLN200 00.0% IN-DB2   151   0 0 ME_PAW
+ 00:00:03.9 SYSLN200 00.0% IN-DB2    66   0 0 db2jcc_appli
```

---
Agenda

- Topology of Distributed Applications
- DB2 Clients
- DB2 Connect
- DB2 z/OS and Distributed Connections
- DB2 z/OS and Connection Resources
- z/OS Workload Manager
- DB2 Private Protocol
- Supplemental Materials
How Many threads can DB2 z/OS handle?
(Is 800 DBATs high?)

\[ CTHREAD + MAXDBAT \leq 2000 \text{ - “well probably not 2000”} \]

- How many threads can be supported? It will be different for each DB2 system based on:
  - Available Virtual Storage
    - DBM1 Address Space (V7 – Bufferpools, Dynamic Statement Cache)
  - Thread footprint (size of the thread)
    - Stored Procedures, Large SQL Statements
  - Duration of a thread
    - Long running program, Commit Frequency, CURSORHOLD
  - Number of allied threads (threads started locally)
    - CAF/RRSAF, TSO QMF Users, Batch Jobs, IBM Regions, CICS transactions, DB2 Utilities and …
  - Number of Remote threads
What happens if you don’t have enough storage for threads? (bad things…)

- Examples of problems:
  - Abends due to either E20003 or E20016 abends of user transactions
  - 878 abends of DBM1 if no storage available for must complete work (backout)

- Key Thread Parameters:
  - CTHREAD - Number of allied threads (threads started at the local)
  - MAXDBAT – Database Access Thread (DBAT)
  - \( CTHREAD + MAXDBAT \leq 2000 \) is the maximum
    - Many customers support MAXDBATs 100s to 400
    - \textit{Note: this is very dependent on site thread profile}

- The MAXDBAT parameter is the limiter to prevent over allocation of DB2 thread memory for DDF connections
Where Do Threads Live in DB2 z/OS – V8

DBM1 64 bit Virtual Memory Map

- User Private area
  - Buffer pools, BM control blocks
  - Castout buffers
  - RID pool
  - EDM DBD cache (OBDs)
  - Global dynamic stmt cache
  - Sort pool
  - Trace tables (Global, Lock, BB)
  - Accounting blocks
  - Compression dictionaries
  - IRLM locks

- Dead Space

- Below 2GB
  - Most thread storage. Local DSC
  - Other EDM and RID component

<= 16EB "beam"(?)

<= 2GB "bar"
How Do I figure out thread usage and virtual storage available?

See:
- John Campbell’s - Virtual Storage Relief in DB2 V8...What to Expect
  ibm.com/software/zseries/telecon/15nov
- Judy Ruby-Brown’s Teleconference:
  Data Sharing Health Checks….What We Have Learned:
  - Section “Virtual Storage in DBM1” reviews thread storage utilization and provides a guide to compute utilization and use of memory tool.
- Memory Reporting Tool:
- If there is a need to have many more DBATs than a single DB2 subsystem can hold, data sharing can help this by scaling a DB2 system wide (with more members).
Estimating Maximum Number of Threads

“Basic” Formula for estimating Number of Active Threads

\[
\text{Working Max} = \text{Extended Region Size} \\
\quad \quad \quad \text{minus 31bit Extended Low Private} \\
\quad \quad \quad \text{minus 200MB (Basic Cushion)}
\]

\[
\text{Fixed Areas} = \text{Total Getmained Storage below the 2GB bar} \\
\quad \quad \quad \text{plus Total Getmained Stack Storage} \\
\quad \quad \quad \text{plus Total Fixed Storage}
\]

\[
\text{Upper Limit Variable} = \text{Working Max} \text{ minus Fixed Areas}
\]

\[
\text{Thread Footprint} = \frac{(\text{Total Variable Storge} \text{ minus Total Agent System Storage})}{(\text{Allied Threads} \text{ plus Current Active DBATs})}
\]

\[
\text{Max. No. of Active Threads} = \frac{\text{Upper Limit Variable}}{\text{Thread Footprint}}
\]
How Many Treads Can DB2 z/OS Handle?

Excel Using MEMU tool statistics.

CTHREAD + MAXDBAT = protected DB2 z/OS

Notice that “AZ” “Thread Footprint” is not the same.

Note: If you have anything other than zero in columns “AK” or “AL” it is a BAD thing!
Therefore listen for:
“…were adding some new web servers…”

- Monitor thread footprint regularly
- Be conservative and don’t over commit DBATs
- If running short on storage remember: CTHREAD and MAXDBAT use the same area
- Avoid WITHHOLD/CURSORHOLD (default for DB2 Clients) cursors:, which causes cursors to be held over commit points, and prevents the DB2 threads from being reused until the application disconnects from the database.
Agenda

- Topology of Distributed Applications
- DB2 Clients
- DB2 Connect
- DB2 z/OS and Distributed Connections
- DB2 z/OS and Connection Resources
- z/OS Workload Manager
- DB2 Private Protocol
- Supplemental Materials
What are we talking about on the “Roadmap”?

1. Web Application server With DB2 Connect or Type-4 Java Driver or IBM Data Server Driver
   - Edge server
   - Web Application server
   - DB2 Connect Gateway server
   - IP Sprayer (Load Balancing)
   - Virtual IP Addressing (VIPA) & SysPlex Distributor

2. DB2 Connect Gateway server on zLinux and Hypersockets
   - zIIP DB2-DF
   - DB2 Connect Gateway server

3. Web Application server
   - Client
   - IBM Client
   - Web Application server
   - IBM Client
   - Web Application server
   - IBM Client
   - Web Application server
   - IBM Client
   - Web Application server
   - IBM Client
Workload Manager Overview

IWEB
DB2
DDF
JES2
OMVS
STC

Service Class
- Critical
- Ad hoc
- Accounting & Payroll
- Marketing & Sales

Report Class
- Marketing
- Sales
- Headquarters
- Test

WLM Rules

business objectives
monitoring
What Workload Manager (WLM) does...

- DB2 does not have a TP monitor of its own and uses WLM control workload management
- WLM is used to manage priority of threads
- Some customers have only one service class for all DDF work so every thread is treated as equal
- Use multiple periods for the DDF service class to change the priority of long running threads:
  - Good to control run away queries
  - Long running batch DDF threads through put will be reduced though
- **WLM is mainly used when a system is very busy. Therefore, if application performance is fine when the system utilization is low and application performance degrades as the CPU gets busy, then the WLM policy probably needs to be adjusted**
Workload Manager (WLM)

Attention: If you do not define classification rules for DDF work and you are running in WLM goal mode, then all your DDF work will run in the SYSOTHER service class, which has a discretionary goal. This means that your DDF work will only run if the system has no other work to do, which normally means that your DDF work is executed at a very low priority.

Before DB2 V4 and MVS/ESA V5.2, all work in the DB2 distributed data facility address space was running with the same dispatching priority. Obviously this was not desirable, since you could not distinguish between your high-priority DDF work and lower-priority DDF work. With the introduction of enclaves, DDF transactions can be managed separately from the DDF address space itself. Work coming into DDF is classified as an assigned service class by workload manager (WLM). Depending on the classification rules you specify, WLM can assign service classes to DDF work, and associate different goals with these service classes according to the type of your DDF work.

When the enclave is deleted depends on whether the DBAT can become pooled, type 2 connections only (DRDA). If the DBAT becomes pooled, the enclave is deleted. If the DBAT cannot become pooled, the enclave is only deleted at thread termination time. Since WLM assigns the performance goals to the enclaves, it is the lifetime of the enclave that WLM takes as the duration of the work. Therefore, when you run with CMTSTAT=INACTIVE, DDF creates one enclave per transaction, and response time goals and multiple time periods can be used.
How do I manage different workloads?  
(AKA – Identifying the loved ones.)

1) Create Service Classes: Rule of thumb 3 groups – High, Medium & Low
2) Identify work: User Id., Program Name, etc..
3) Assign service class to identified work
4) Service Classes can be subdivided into Reporting Classes for more detailed reporting information
5) Have several periods in a service class
6) Each service class’s first period can be at a high priority for a sort time (Don’t penalize short duration work)
Service Classes

Service classes identify the priority of different types of work

<table>
<thead>
<tr>
<th>Service Class Selection List</th>
<th>Row 1 to 11 of 11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>Menu Bar</td>
</tr>
</tbody>
</table>

**Action Codes:**
1 = Create, 2 = Copy, 3 = Modify, 4 = Browse, 5 = Print, 6 = Delete

<table>
<thead>
<tr>
<th>Action</th>
<th>Class</th>
<th>Description</th>
<th>Workload</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ASCH</td>
<td>APPC Transaction Programs</td>
<td>STCSTASKS</td>
</tr>
<tr>
<td>2</td>
<td>BATCH</td>
<td>Batch Workload</td>
<td>BATCH</td>
</tr>
<tr>
<td>3</td>
<td>CICS</td>
<td>CICS Transactions</td>
<td>ONLINE</td>
</tr>
<tr>
<td>4</td>
<td>DB2QUERY</td>
<td>DB2 Sysplex Queries</td>
<td>DATABASE</td>
</tr>
<tr>
<td>5</td>
<td>DDFDEF</td>
<td>DDF Default Requests</td>
<td>DATABASE</td>
</tr>
<tr>
<td>6</td>
<td>DDFBAT</td>
<td>DDF low priority</td>
<td>DATABASE</td>
</tr>
<tr>
<td></td>
<td>DDFONL</td>
<td>DDF High priority</td>
<td>DATABASE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Service class</th>
<th>Period</th>
<th>Duration</th>
<th>Importance</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDFONL</td>
<td>1</td>
<td>500</td>
<td>2</td>
<td>90% within 1 second</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>500</td>
<td>3</td>
<td>Velocity 40</td>
</tr>
<tr>
<td>DDFDEF</td>
<td>1</td>
<td>500</td>
<td>3</td>
<td>80% within 2.5 seconds</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>500</td>
<td>4</td>
<td>Velocity 20</td>
</tr>
<tr>
<td>DDFBAT</td>
<td>1</td>
<td>500</td>
<td>4</td>
<td>80% within 4 seconds</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>500</td>
<td>5</td>
<td>Velocity 10</td>
</tr>
</tbody>
</table>
Workload Manager…

50 Concurrent Users + Killer Queries

Runaway queries cannot monopolize system resources
aged to low priority class

<table>
<thead>
<tr>
<th>period</th>
<th>duration</th>
<th>velocity</th>
<th>importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5000</td>
<td>80</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>50000</td>
<td>60</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1000000</td>
<td>40</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>10000000</td>
<td>30</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>discretionary</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Classification Rules

Classification rules connect Service Classes with work qualifiers, or DB2 threads/transactions. When setting up classification rules, plan for “defaults” to accommodate new unplanned for production applications.

The Service Class performance data can be reported in RMF which can provide information on z/OS system level thread activities. Reporting Classes can be created in addition to Service Classes for details within Service Classes.
Also: Managing Dynamic SQL - RLF/Governor

- Resource Limit Facility – DB2 9 for z/OS Enhancements
  - RLF will be enhanced to allow CPU cost to be controlled based on:
    - Client workstation name
    - Client application name
    - Client userid
    - IP address
  - There is a new RLF table: authid.DSNRLMTnn
    - RLF will start if either (or both) DSNRLSTnn or DSNRLMTnn exists
    - Created in job DSNTIJSG for installation or migration
  - Allows fine-grain RLF limits for apps that exploit the Set Client Information APIs (SAP, PeopleSoft, Siebel, WebSphere)
Sample DDF Definition – Available on the web

Work Qualifiers

Work qualifiers are used to help identify a thread or work

- AI - Accounting Information*
- CI - Correlation Information*
- CN - Collection Name
- CT - Connection Type
- CTG - Connection Type Group
- LU - LU Name
- LUG - LU Name Group
- NET - Net ID
- NETG - Net ID Group
- PC - Process Name
- PF - Perform
  * Popular choices
  ** pureQuery can set this
- PFG - Perform Group
- PK - Package Name **
- PKG - Package Name Group
- PN - Plan Name
- PNG - Plan Name Group
- PR - Procedure Name
- PX - Sysplex Name
- SI - Subsystem Instance
- SIG - Subsystem Instance Group
- SSC - Subsystem Collection
- UI - Userid
Providing Workload Classification Attributes from the Client

- ADO.Net Examples – use DB2Connection properties:
  - `myConnection.ClientUser = "PAWUSER"
  - `myConnection.ClientApplicationInformation = "PAWVSAPPL"

- Universal Driver for Java applications (JCC T2 or T4) - use methods against connection class instance
  - `setClientUser`
  - `setClientApplicationInformation`,
  - `setClientWorkStation`
  - `setClientAccountingInformation`
Adding Java clientProgramName will set Correlation Id.
.Net setting WLM parameters

```vbnet
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim myConnString As String = "Database=NCDB200;UserID=dba031;Password=xxxx;Server=denom";
    Dim myConnection As DB2Connection = New DB2Connection(myConnString)

    'Can set user, workstation and etc before or after opening the connection
    myConnection.Open()
    myConnection.ClientUser = "PAWUSER"
    myConnection.ClientApplicationInformation = "PAWVSAPPL"
    myConnection.Open()

    Dim mySelectQuery As String = "SELECT empno,firstname.lastname,workdept from dba031.employees"
    Dim myCommand As New DB2Command(mySelectQuery, myConnection)
```

<table>
<thead>
<tr>
<th>Trans Program Name:</th>
<th>DB2 Information Management Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBA031</td>
<td>SYSSH200</td>
</tr>
<tr>
<td>DB2SERV</td>
<td>NULLID</td>
</tr>
<tr>
<td>DB2_ORDA</td>
<td>DB2S</td>
</tr>
<tr>
<td>SQL9015NT</td>
<td>PAWUSER</td>
</tr>
<tr>
<td>PAWVSAPPL</td>
<td>DSNSG</td>
</tr>
</tbody>
</table>
What are packages and why / when do I need to rebind?

http://www.ibm.com/developerworks/wikis/display/DB2/DB2+and+.NET+FAQs#DB2and.NETFAQs-ChangesinCLI%2F.NETandJDBCpackages

- Packages (in DB2 terms) are containers for the necessary instructions to execute a SQL statement. They are actually stored in the database itself. This section describes only the packages necessary for the dynamic programming languages (ODBC, .NET and JDBC), additional packages exist for other utilities and are outside the scope of this article.

- For dynamic languages (like .NET, ODBC and JDBC), DB2 has a set of "empty" packages stored in the database, and the client utilizes those packages as the application executes SQL statements. The packages used for these dynamic languages have been standardized with V8 of our client drivers.

- These packages need to exist on the database before any dynamic statements (from ODBC, .NET and JDBC) can be used. There is no automatic bind process from the client, they must be explicitly bound. For DB2 LUW servers, the packages are created at database creation time by the server.

- There is one other package - SYSSTAT, this package contains misc static functions that can not be executed dynamically.
# Dynamic Package Names

<table>
<thead>
<tr>
<th>Bind file Name</th>
<th>Package name</th>
<th>Needed by DB2 servers on Linux, UNIX, and Windows</th>
<th>Needed by Host servers</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>db2clipk.bnd</td>
<td>SYSSHxxy</td>
<td>Yes</td>
<td>Yes</td>
<td>dynamic placeholders - small package WITH HOLD</td>
</tr>
<tr>
<td></td>
<td>SYSSNxyy</td>
<td>Yes</td>
<td>Yes</td>
<td>dynamic placeholders - small Package NOT WITH HOLD</td>
</tr>
<tr>
<td></td>
<td>SYSLHxxy</td>
<td>Yes</td>
<td>Yes</td>
<td>dynamic placeholders - large package WITH HOLD</td>
</tr>
<tr>
<td></td>
<td>SYSLNxyy</td>
<td>Yes</td>
<td>Yes</td>
<td>dynamic placeholders - large package NOT WITH HOLD</td>
</tr>
<tr>
<td>db2clist.bnd</td>
<td>SYSSTAT</td>
<td>Yes</td>
<td>Yes</td>
<td>common static CLI functions</td>
</tr>
<tr>
<td>db2schema.bnd</td>
<td>SQLL9vyy</td>
<td>Yes</td>
<td>No</td>
<td>catalog function support</td>
</tr>
</tbody>
</table>

**Note:**
- ’S’ represents a small package and ’L’ represents a large package
- ’H’ represents WITH HOLD, and ’N’ represents NOT WITH HOLD.
- ’v’ represents the DB2 server version: for example, E=Version 8, F=Version 9
- ’x’ is the isolation level: 0=NC, 1=UR, 2=CS, 3=RS, 4=RR
- ’yy’ is the package iteration 00 through FF
- ’zz’ is unique for each platform

For example, for the dynamic packages:
- SYSSN100 A small package (65 sections) where all cursor declarations are for non-held cursors. Bound with isolation level UR. This is the first iteration of that package.
- SYSLH401 A large package (385 sections) where all cursor declarations are for held cursors. Bound with isolation level RS. This is the second iteration of that package.

Previous versions of DB2 servers do not need all of the bind files and will therefore return errors at bind time. Use the bind option SQLERROR(CONTINUE) so that the same package can be bound on all platforms and errors will be ignored for any statements not supported there.
Why are there so many packages, and what in the world does “package iteration 00 through FF” mean???

Used to provide statement handles and provides concurrency for execution of SQL statements. Large packages (SYSL…) can have more that 3 iterations. An HY014 SQLSTATE may be returned on the call to SQLPrepare(), SQLExecute(), or SQLExecDirect() if this limit is exceeded. Normal default binding provides 3 iterations.

SQL0805N error when performing a DML statement on database.

Question
SQL0805N error, Package.Nullid.SYSLH203 was not found, SQLSTATE=51002, returned when attempting to perform a DML statement on the database server from a remote client application.

Cause
The package, SYSLH203, doesn't exist on the database server.

Answer
Depending on the type of statement you are executing, DB2 will use a particular package on the server. By default, DB2 creates three packages for each type of package. In this case NULLID.SYSLH2yy is reserved for statements with CURSORSTABLE and isolation level Cursor Stability. The package SYSLH203 means that DB2 is looking for the 4th package (200 is first, 201 is second, etc) of this type, but it does not exist. You can create more packages on the server by connecting to the database and issuing the following bind command from the /sqlib/bnd directory:
db2 bind @db2cli.lst blocking all grant public sqlerror continue CLIPKG 5
Note: CLIPKG 5 will create 5 large packages, and will give you the package that your application is looking for, as well as one more in this case.

Related information
CLIPKG CLI/ODBC Configuration Keyword
CLI bind files and package names
BIND

http://www-01.ibm.com/support/docview.wss?uid=swg21208123
Example: DDF Service Class Information

<table>
<thead>
<tr>
<th>Period Number</th>
<th>Importance</th>
<th>Duration in Service Units</th>
<th>Percentile</th>
<th>Response Time Goal or Velocity</th>
<th>Goal Description</th>
<th>Percentile Response Time Goal</th>
<th>Velocity Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>500 MilliSecs</td>
<td>80%</td>
<td>2500 MilliSecs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Ouch!
My importance is only 4 and I'm using the DDFDEF service class
How can I figure out thread Enclave activity?

RMF Monitor III
- Overview
- Enclave Report
zIIP Eligible DB2 z/OS V8 Workloads

- ERP, CRM and multi-tier application serving*
  - For applications, running on z/OS, UNIX, Linux, Intel, or Linux on System z, that access DB2 for z/OS V8 on a System z9, via DRDA over a TCP/IP connection DB2 gives z/OS the necessary information to have portions of these SQL requests directed to the zIIP.

- Data warehousing applications*
  - Requests that utilize DB2 for z/OS V8 star schema parallel queries may have portions of these SQL requests directed to the zIIP when DB2 gives z/OS the necessary information.
    - This function has been enhanced to include all parallel queries.

- Some DB2 for z/OS V8 utilities*
  - A portion of DB2 utility functions used to maintain index maintenance structures (example LOAD, REORG, and REBUILD INDEX) typically run during batch, can be redirected to zIIP.

- The zIIP is designed so a program can work with z/OS to have a portion of it’s Service Request Block (SRB) enclave work directed to the zIIP. The above types of DB2 V8 work are those executing in SRB enclaves, portions of which can be sent to the zIIP.
DB2 z/OS V9 Continues to add zIIP features –
(DRDA invoked SQL-PL stored procedures zIIP eligible)

Native SQL Procedural Language

- Eliminates generated C code and compilation
- Fully integrated into the DB2 engine
- Extensive support for versioning:
  - VERSION keyword on CREATE PROCEDURE
  - CURRENT ROUTINE VERSION special register
  - ALTER ADD VERSION
  - ALTER REPLACE VERSION
  - ALTER ACTIVATE VERSION
- BIND PACKAGE with new DEPLOY keyword

See Roger Miller’s May 26, 2006 web cast for more DB2 z/OS V9 details
Example of RMF data for zIIP with DRDA

### DRDA Run

2 zIIPs – Exercise zIIP reduced ‘Needs Help Dispatcher’

```
CPU ACTIVITY

<table>
<thead>
<tr>
<th></th>
<th>z/OS VIR7</th>
<th>SYSTEM ID SYSC</th>
<th>START 07/07/20</th>
<th>END 07/07/20</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>2094</td>
<td>MODEL 750</td>
<td>H/W MODEL S54</td>
<td></td>
</tr>
<tr>
<td>---CPU---</td>
<td>ONLINE TIME</td>
<td>LPAR BUSY</td>
<td>MVS BUSY</td>
<td>CPU SERIAL</td>
</tr>
<tr>
<td>0 CP</td>
<td>100.00</td>
<td>63.76</td>
<td>63.76</td>
<td>07B10E</td>
</tr>
<tr>
<td>1 CP</td>
<td>100.00</td>
<td>57.48</td>
<td>57.48</td>
<td>07B10E</td>
</tr>
<tr>
<td>2 CP</td>
<td>100.00</td>
<td>60.67</td>
<td>60.67</td>
<td>07B10E</td>
</tr>
<tr>
<td>3 CP</td>
<td>100.00</td>
<td>40.95</td>
<td>40.95</td>
<td>07B10E</td>
</tr>
<tr>
<td>CPU TOTAL/AVERAGE</td>
<td>55.72</td>
<td>55.72</td>
<td>07B10E</td>
<td>15710</td>
</tr>
<tr>
<td>4 IIP</td>
<td>100.00</td>
<td>81.11</td>
<td>81.11</td>
<td>07B10E</td>
</tr>
<tr>
<td>7 IIP</td>
<td>100.00</td>
<td>88.52</td>
<td>88.52</td>
<td>07B10E</td>
</tr>
<tr>
<td>IIP AVERAGE</td>
<td>84.81</td>
<td>84.81</td>
<td>07B10E</td>
<td></td>
</tr>
</tbody>
</table>
```

**REPORT BY:** POLICY=WIMPOL  WORKLOAD=WAS_NKL  SERVICE CLASS=CISD0F  RESOURCE GROUP=*NONE  CRITICAL =NONE

```
TRANSACTIONS  TRANS-TIME HHH.MM.SS.TTT  --DASD I/O--  ---SERVICE----  SERVICE TIMES  ---APPL %---
AVG  37.87  ACTUAL  19  SSSHRT 11070  TOC  0  CPU  1895.4  CP  160.73 
MPL  37.87  EXECUTION  19  RESP  2.2  CPU  50884K  SRB  0.0  AAPCP  0.00 
ENDED 1149619  QUEUED  0  CONN  0.3  MSO  0  RCIT  0.0  IIPCP  15.57 
END/s  1916.04  R/s AFFIN  0  DISC  1.9  SRB  0  IIT  0.0 
#SNAPS  0  INELIGIBLE  0  Q*PEND  0.2  TOT  50884K  MST  0.0  AAP  0.00 
EXCTD  0  CONVERSION  0  IOSQ  0.0 /SEC  84807  AAP  0.0  IIP  155.18 
AVG ENC  37.87  STD DEV  122  IIP  931.1 
```

The RMF Service class reports can provide information on thread activities too. Reporting Classes can be created in addition to Service Classes for details within Service Classes.
Comparison of External (WLM) and Native Stored Procedures

![External Throughput Ratio](image1)

![Internal Throughput Ratio](image2)

Redbook DB2 9 for z/OS Performance Topics SG24-7473
Comparison of External and Native Stored Procedures - Findings

The comparison of the three workloads had the following results:

- **For the simple workload**
  - A 19% improvement in external throughput rate (ETR) with DB2 V9 native SQL procedures
  - An 11% improvement in ITR with DB2 V9 native SQL procedures

- **For the Relational Warehouse Workload**
  - A 27% improvement in ETR with DB2 V9 native SQL procedures
  - A 37% improvement in ITR with DB2 V9 native SQL procedures
  - zIIP total redirect eligibility (IIP+IIPCP) increased from 8% to 55%

In the SQL procedure support in DB2 V8, there are cases where SQL functions are converted into SELECT statements when the original SQL procedure source is precompiled. The same SQL functions are converted into SET statements in the DB2 V9 native SQL language support. This occurs in two of the Relational Warehouse Workload transactions and creates the higher performance benefit that is seen.

- **For the financial workload**
  - Equivalent ETR between DB2 V9 native and V8 external SQL procedures
  - An 18% improvement in ITR with DB2 V9 native SQL procedures
  - zIIP total redirect eligibility (IIP+IIPCP) increased from 6% to 50%
More Quick Hits …

- Stored Procedures - Performance of different languages …

<table>
<thead>
<tr>
<th>Language</th>
<th>Base Billable Cost</th>
<th>Billable Cost after zIIP and/or zAAP acceleration</th>
</tr>
</thead>
<tbody>
<tr>
<td>COBOL stored proc</td>
<td>1X (BASE)</td>
<td>.88x</td>
</tr>
<tr>
<td>C stored proc</td>
<td>.95x</td>
<td>.83x</td>
</tr>
<tr>
<td>Remote SQLJ</td>
<td>1.78x</td>
<td>1.06x</td>
</tr>
<tr>
<td>SQLJ stored proc</td>
<td>1.21x</td>
<td>1.15x (zIIP + zAAP)</td>
</tr>
<tr>
<td>JDBC stored proc</td>
<td>2.11x</td>
<td>1.76x (zIIP + zAAP)</td>
</tr>
<tr>
<td>External SQL stored proc</td>
<td>1.62x</td>
<td>1.49x</td>
</tr>
<tr>
<td>Native SQL stored proc</td>
<td>1.14x</td>
<td>.65x</td>
</tr>
</tbody>
</table>
Agenda

- Topology of Distributed Applications
- DB2 Clients
- DB2 Connect
- DB2 z/OS and Distributed Connections
- DB2 z/OS and Connection Resources
- z/OS Workload Manager
- DB2 Private Protocol
- Supplemental Materials
Private Protocol

- **Inbound/outbound private protocol still works**
  - V5, V6, V8 and V9 officially deprecated private protocol
- **DBPROTCL Z Parm has been removed from V9**
  - DBPROTOCOL(DRDA) will be assumed for any BIND/REBIND PLAN/PACKAGE where DBPROTOCOL option is not specified
  - If DBPROTOCOL(PRIVATE) explicitly specified, DSNT226I message will be issued and BIND will complete with a warning return code (4)
  - Therefore, in V9 you cannot have a default DB Protocol of Private anymore
- **DB2 10 for z/OS does not support Private Protocol**
Private protocol to DRDA protocol Catalog Analysis Tool (DSNTP2DP)

- Tool provided in DB2 V9
- DB2 REXX support must be enabled
- APAR PK40433 will permit program to be run against V7 and V8 subsystems
- APAR PK44544 (not yet closed) will change DSNTIJPD (sample job to run tool)
  - Include optional step to create additional indexes on catalog to improve tool’s performance
  - Include optional step to REBUILD INDEX with inline statistics on indexes and RUNSTATS on index that cannot have inline statistics during rebuild
  - Include optional step to delete indexes
- DB2 for z/OS Examples Trading Post allows V7/V8 subsystem user to download tool (with no service provided) and JCL
- Program currently will only generate commands to migrate plans or packages which can be determined to have a remote location dependency
  - Plans or packages which utilize embedded dynamic SQL programming will usually not have a remote location dependency that can be determined by querying the catalog
Private Protocol Considerations

- Since DB2 V6, you can use aliases and three-part names in combination with the DRDA protocol. This provides some form of location transparency and makes migration from DB2 PP to DRDA easier.
- Don’t forget remote binds of plans or packages could increase the size of remote catalogs – not a bad thing just think about it.
Functions not available in the DB2 Private Protocol

- Connections using TCP/IP
- Connections to RDBMS other than DB2 for z/OS and OS/390
- Connection using the SQL CONNECT statement
- DDL (for example, CREATE, DROP) and DCL (for example, GRANT, REVOKE) SQL statements
- Stored procedures
- Remote bind
- Static SQL
- User-defined and LOB data type
- Thread pooling
- Any new features new in V7, V8 or V9
Agenda

- Topology of Distributed Applications
- DB2 Clients
- DB2 Connect
- DB2 z/OS and Distributed Connections
- DB2 z/OS and Connection Resources
- z/OS Workload Manager
- DB2 Private Protocol
- Supplemental Materials
Additional Information

This is the BOOK to have!
Available at: ibm.com/redbooks
SG24-6952-01

Too bad they don’t have
“DB2 Connect” in the title…

www.redbooks.ibm.com/abstracts/sg246952.html
Additional Information:

- Redbook – DB2 9 for z/OS Distributed Functions SG24-6952-01
- John Campbell’s: 2006 IOD Conference Presentation 1282 – DB2 for z/OS V8 Migration and Experiences
- Melanie Stopfer’s: 2007 IOD Conference Presentation 1218 – DB2 Connect to DB2 z/OS Problem Determination & Performance Tips & Tricks
- John Campbell’s - Virtual Storage Relief in DB2 V8...What to Expect http://www.ibm.com/software/zseries/telecon/15nov
- Leon Kats nelson DB2 Technical Conference 2004 - DB2 Connect: Best Practices
- Thanks to: Jim Pickel, Maggie Lin, and Brent Gross: DB2 DRDA and .Net Development
The End!