

Melanie Stopfer & Enzo Cialini

 mstopfer1

Upgrading to DB2 LUW 11.1 Best Practices



© 2016 IBM Corporation

Melanie Stopfer is a IBM Consulting Learning Specialist and Developer and recognized worldwide as a DB2, IBM BLU and Analytics subject matter expert. She has provided in-depth technical customer support in recovery, performance and database upgrade best practices since 1988. In 2009, Melanie was the first DB2 LUW speaker to be inducted into the IDUG Speaker Hall of Fame, selected Best Overall Speaker a record breaking seven times (IDUG EMEA 2009, 2011, 2012, 2015; IDUG NA 2009, 2012, 2014), and included in Top 10 Overall Speakers in all years presenting. She has successfully presented at all IBM IOD and Insight conferences and been recognized as a top presenter. Please connect with Melanie on Twitter at [@mstopfer1](https://twitter.com/mstopfer1) and LinkedIn.

Thank you to Enzo Cialini for formatting my previous presentation into the IBM Analytics format, starting the upgrade, and motivating me to complete the DB2 11.1 Upgrade Best Practices presentation.

Safe Harbor Statement

Copyright © IBM Corporation 2016. All rights reserved.

U.S. Government Users Restricted Rights - Use, duplication, or disclosure restricted by GSA ADP Schedule Contract with IBM Corporation

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON CURRENT THINKING REGARDING TRENDS AND DIRECTIONS, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. FUNCTION DESCRIBED HEREIN MAY NEVER BE DELIVERED BY IBM. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

IBM, the IBM logo, ibm.com and DB2 are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

© IBM Corporation 2016. All Rights Reserved.

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

IBM, and the IBM logo, are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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

Announcing NEW Podcasts - Origin Stories from DB2 Community!



<http://createdatabase.fm>

 [@_createdb](https://twitter.com/_createdb)

URL for podcast episodes:

Melanie Stopfer
<http://createdatabase.fm/episodes/1>

Fred Sobotka
<http://createdatabase.fm/episodes/2>

Adam Storm
<http://createdatabase.fm/episodes/3>

Iqbal Goralwalla
<http://createdatabase.fm/episodes/4>

IDUG CONTENT BLOG

- [DB2 for LUW V11.1 : Upgrade and Recovery Enhancements](#)
By: Melanie Stopfer Sun Jul 3rd, 2016
<http://www.idug.org/p/bl/ar/blogaid=510>
- [DB2 for LUW V11.1 BLU Performance](#)
By: Joe Geller Wed Jun 29th, 2016
- [Scaling DB2 BLU Horizontally with DB2 for LUW V11.1](#)
By Joe Geller and Ian Bjorhovde Thu Jun 23rd, 2016
- [Try out dsmtop \(an alternative to db2top\)](#)
By: Paul Bird Tue Jan 26th, 2016
- Go to IDUG CONTENT BLOG:
<http://www.idug.org/p/bl/et/blogid=278>

For more information see IDUG CONTENT BLOG at www.idug.org.

Objectives

- Describe changes to DB2 11.1 packaging and software and operating system requirements.
- Understand DB2 11.1 changes to DDL, utilities, tools and monitoring.
- Configure a database and instance to implement DB2 11.1 new features.
- Analyze DB2 11.1 changes that affect applications and database runtime behavior.
- Develop a DB2 11.1 upgrade strategy



Abstract:

Are you responsible for upgrading your database to the latest version of DB2 LUW? If so, don't miss this presentation! Melanie is digging into DB2 11.1 upgrade best practices details! If you need a detailed step-by-step approach to implement the newest version of DB2 11.1, come learn about often overlooked but very important best practices to understand before and after upgrading. Learn how to take advantage of the new DB2 11.1 great functionality. Unlock the hidden secrets in DB2 11.1. Melanie will walk you through a cookbook approach to upgrading to DB2 11.1 without stopping HADR and reinitializing your HADR standby(s). She will also give you a step-by-step approach to recovering through a DB2 11.1 server upgrade and to rolling forward through database version upgrades.

DB2 11.1 Upgrade Enhancement Overview

- **Upgrade directly from 9.7 instead of having to go through another version, such as 10.1 or 10.5.**
 - **Upgrade is now a recoverable operation**
 - Recovery procedure involving roll-forward through database version upgrades for single-partition DB2 and DB2 pureScale upgrading from DB2 10.5 Fix Pack 7 or later.
- 
- **Avoid need for offline backup image during upgrade procedure**
 - No longer required to perform offline database backup before or after upgrade
 - **HADR can now be upgraded without need to re-initialize standby database after perform upgrade on primary database.**
 - Only single-partition DB2 upgrading from DB2 10.5 Fix Pack 7 or later
 - Reduces window where no standby database exists
 - Eliminates cost of sending backup image to standby site for re-initialization.
 - Re-initialization of standby is still option, but not recommended option.

DB2 11.1 enables users to easily upgrade to the latest DB2 database product. For example, users can now upgrade directly from V9.7 instead of having to go through another version, such as 10.1 or 10.5.

DB2 11.1 introduces the ability to roll-forward through database version upgrades for single-partition DB2 Enterprise Server Edition and DB2 pureScale users upgrading from DB2 10.5 Fix Pack 7, or later. For such configurations, users are no longer required to perform an offline backup of existing databases before or after they upgrade, as a recovery procedure involving roll-forward through database upgrade now exists.

Roll forward recovery through database upgrade is now supported. This includes the ability for an HADR standby to replay through log data written and shipped by the primary database. Although database upgrade is now recoverable in all configurations, only databases upgrading from DB2 10.5 Fix Pack 7 or later in single partition Enterprise Server Edition and pureScale configurations can take advantage of the new recoverability support. In configurations using multiple partitions, roll forward through a database upgrade is not supported. Roll forward through a database upgrade is also not supported for databases upgrading from DB2 Version 9.7, 10.1 and DB2 Version 10.5 Fix Pack 6 or earlier.

Database upgrade is now a recoverable event which means it is possible, depending on your database environment, to avoid the need to take an offline database backup before and after the database upgrade. For DB2 Version 10.5 Fix Pack 7 or later databases in configurations using single-partition Enterprise Server Edition or pureScale, the ability to roll forward through a database upgrade is supported. In configurations using multiple partitions, roll forward through a database upgrade is not supported. Roll forward through a database upgrade is also not supported for databases upgrading from DB2 Version 9.7, 10.1 and DB2 Version 10.5 Fix Pack 6 or earlier.

For single-partition DB2 Enterprise Server Edition users upgrading from DB2 10.5 Fix Pack 7, or later, high availability disaster recovery (HADR) environments can now be upgraded without the need to re-initialize the standby database after performing an upgrade on the primary database.

Only for single-partitioned DB2 Enterprise Server Edition users, the upgrade of a HADR pair from a supported release and fix pack will no longer need the re-initialization of the standby through the use of a backup image

shipped from the primary after upgrade. Re-initialization of the standby is still an option if the user wishes, but is no longer the recommended option.

The supported HADR upgrade procedure requires the primary and standby databases to both be, at a minimum, at DB2 10.5 Fix Pack 7 level. Any other prior level is not supported and attempts to use the documented procedure will fail. For these prior level databases, the standby will still have to be re-initialized by a backup image shipped from the primary after upgrade.

Upgrading to DB2 11.1

- **DB2 11.1 continues the most versatile hybrid Database Software**

Component	Upgrade Required?
Data Server	Yes, instance & database
Client	Optional

- **If have DB2 9.7, 10.1, or 10.5 copy installed and want to use DB2 11.1, need to upgrade to 11.1.**
 - DB2 11.1 is new version.
 - Cannot apply fix pack to upgrade from prior DB2 version to DB2 11.1.
- **Upgrading not required if only installing fix packs**
- **Applications and routines may require changes (rare)**

DB2 11.1 for Linux, UNIX, and Windows offers database efficiency, simplicity, and reliability in the form of new features and enhancements that address a wide variety of business requirements. Comprehensive enterprise security, simplified installation and deployment, better usability and adoption, a streamlined upgrade process, enhancements to very large databases, and vast improvements to BLU Acceleration are the key benefits that this technology delivers.

Version upgrades are required for the Data Server component which means instance and associated databases. However, the Client components are optional and not required to be upgraded to Version 11.1.

If you have a DB2 V9.7, 10.1, or 10.5 copy already installed and you want to use DB2 11.1 instead, you need to upgrade to 11.1. DB2 11.1 is a new version. You cannot apply a fix pack to upgrade from a prior version to 11.1.

We do not force an upgrade on you via a fix pack. This is really important as you can be confident that when you apply a fix pack, you are not surprised.

Recommended Reading

- **DB2 Upgrade Portal**
 - One-stop-shop for essential information
<http://www.ibm.com/software/data/db2/upgrade/portal>
- **DB2 Upgrade Home Page**
 - Links to webcasts, videos, wikis, etc.
<http://www.ibm.com/software/data/db2/upgrade>
- **DB2 V11.1 Knowledge Center**
 - **DB2 11.1 Release Notes**
 - **DB2 11.1 What's New**
 - <http://ibm.co/29b7q1B>
- **Download DB2 V11.1 trials**
 - <http://bit.ly/DB2v11Web>
- **DB2 V11.1 Data Sheet**
 - <https://ibm.biz/Bd4eBq>
- **DB2 on Cloud**
 - <http://www.ibm.com/analytics/us/en/technology/cloud-data-services/db2-on-cloud/index.html>



Here you see a few items which are very useful for you for your planning purposes before, during, and even after your upgrade.

These are resources that will help you feel confident in this process and help you succeed

Planning Your Upgrade to DB2 LUW 11.1

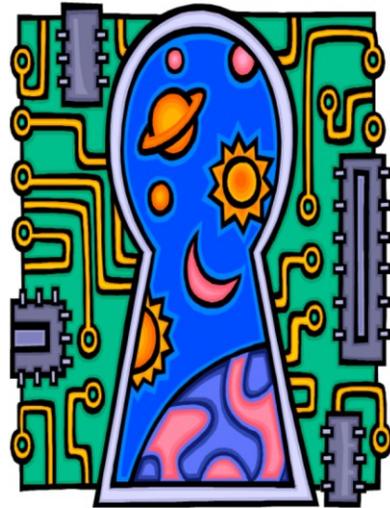
DB2 11.1 LUW Upgrade from *pre-DB2 Version 11.1* copy:

- **DB2 10.5-> DB2 11.1**
- **DB2 10.1-> DB2 11.1**
- **DB2 9.7 -> DB2 11.1**

- **If DB2 servers are running on release prior DB2 9.7:**
 - Upgrade to DB2 9.7, 10.1 or 10.5 latest fix pack
 - Then upgrade to DB2 11.1

See notes for more details/URLs:

- **DB2 Upgrade Portal**
- **DB2 11.1 English Manuals**
- **DB2 11.1 Translated Manuals**



DB2 11.1 for Linux, UNIX and Windows planned availability for electronic download is June 15, 2016.

In the upgrade tasks, the term *pre-DB2 Version 11.1 releases* refers to DB2 Version 10.5, DB2 Version 10.1 or DB2 Version 9.7.

Refer to the DB2 11.1 Knowledge Center for details about the installation and upgrade process. It is highly recommended that you consult the DB2 Knowledge Center while planning and executing an upgrade. You should also test the upgrade process on non-production servers first. Your first point of reference to learn how to upgrade each component of your DB2 environment and how to create a upgrade plan is the DB2 Knowledge Center. You may also download the DB2 Version 11.1 for Linux, UNIX, and Windows English or translated manuals from the Knowledge Center home page.

Checking Whether Upgrade is Possible

Direct upgrades are supported from Version 9.7 and higher systems. Upgrading your DB2 server requires that you install a DB2 11.1 copy and then upgrade all the instances and databases to be able to run them under the DB2 11.1 copy. You can directly upgrade existing DB2 9.7, 10.1 or DB2 10.5 instances and databases to DB2 11.1. On Windows operating systems, you have an option to automatically upgrade an existing pre-11.1 DB2 copy. If you choose to upgrade your existing DB2 copy during installation, you only need to upgrade your databases after installation.

Review also the DB2 upgrade portal at www.ibm.com/support (formerly known as DB2 migration portal) that provides access to additional resources and up-to-date information about the upgrade process as they become available. These resources include educational material, white papers, and webcasts for upgrade. **The DB2 Upgrade Portal is located at <http://www-01.ibm.com/support/docview.wss?uid=swg21200005>**

Announcement Summary - EOM and EOS dates

- **DB2 11.1 - Announced April 12, 2016 with eGA June 15, 2016**
- **End of Marketing for DB2 10.5 - September 30, 2016**
- **End of Service for DB2 9.7 and DB2 10.1 - September 30, 2017**

DB2 Version	SAP specific EOS for all DB2 Customers
10.5	2025-12-31
10.1	2022-12-31
9.7	2022-12-31
9.5	2017-12-31
9.1	2017-12-31

Both IBM and SAP have agreed to align EOS dates for SAP DB2 LUW clients regardless of licensing, ASL or with IBM contract. Once published IBM EOS for DB2 is reached, both partners will continue to support the DB2 version in alignment with SAP software releases according to above table. IBM direct clients will report their SAP DB2 problems as usual, to SAP Support, also for DB2 after EOS. Call-in to IBM support is not allowed after IBM EOS.

Operating System Support for Database Server

OS	Software	Hardware
AIX	<ul style="list-style-type: none"> AIX Version 7.1 TL3 SP5+ (64-bit AIX kernel required) 	<ul style="list-style-type: none"> 64-bit Common Hardware Reference Platform (CHRP) architecture All processors that are capable of running the supported AIX operating systems
Linux	<ul style="list-style-type: none"> Red Hat Enterprise Linux (RHEL) 6.7+ Red Hat Enterprise Linux (RHEL) 7.1+ SUSE Linux Enterprise Server (SLES) 11SP4+ SUSE Linux Enterprise Server (SLES) 12 CentOS 6.7, 7.1 Ubuntu 14.04 LTS 	<ul style="list-style-type: none"> x86 (Intel® Pentium®, Intel Xeon®, and AMD) 64-bit Intel and AMD processors x64 (64-bit AMD64 and Intel EM64T processors)
pLinux (LE)	<ul style="list-style-type: none"> Red Hat Enterprise Linux (RHEL) 7.1+ SUSE Linux Enterprise Server (SLES) 12 Ubuntu 14.04 LTS 	<ul style="list-style-type: none"> 64-bit Common Hardware Reference Platform (CHRP) architecture POWER® (IBM® eServer™ OpenPower®, iSeries®, pSeries®, System i®, System p®, and POWER Systems that support Linux)
zLinux	<ul style="list-style-type: none"> Red Hat Enterprise Linux (RHEL) 7.1+ SUSE Linux Enterprise Server (SLES) 12 	<ul style="list-style-type: none"> eServer System z®, System z10®, System z13®
Windows	<ul style="list-style-type: none"> Windows 7 (64-bit) Windows 8.1 (64-bit) Windows 10 (64-bit) Windows 2012 R2 (64-bit) 	<ul style="list-style-type: none"> All Intel® and AMD processors capable of running the supported Windows operating systems (64-bit based systems)
<p>For most recent detailed requirements see DB2 11.1 Knowledge Center.</p> <p>Specific OS patches are also in the requirements, review and apply these prior to installing new release!</p>		

This is a list of what is and what is not supported. As this changes and is upgraded frequently the best source is the URL listed at the bottom of the slide.

A simple but important step is to remember to apply all necessary OS specific patches prior to installation. Sometimes much grief and time in problem determination can be avoided if this is performed first!

The following are limitations for Ubuntu 16.04 on zLinux in DB2 11.1:

- No Automated High Availability fail over (TSA)
- No Fault Monitor when using system

The following are pre-requisites for Ubuntu 16.04 on zLinux in DB2 v11.1 GA (fixpack 0):

- xIC Runtime (libxlc-1.2.0.0-160121.s390x.rpm)
- Linux kernel AIO access library - shared library (libaio1)
- Strings utility as part of GNU Binary utilities (binutils)

You can bypass an install that failed during db2prereqcheck due to these system pre-requisites by using the force sysreq install option: **db2setup [other options] -f sysreq**

For libaio1 and binutils one can install the prerequisites utilizing apt-get.

- **sudo apt-get install libaio1**
- **sudo apt-get install binutils**

For xLC, refer to <http://www-01.ibm.com/support/docview.wss?uid=swg21688903> to retrieve a xLC runtime rpm for Linux on Z.

Operating systems no longer supported

In DB2 11.1, the following operating systems and Linux distributions are no longer supported:

- Red Hat Enterprise Linux 5.6
- Ubuntu 10.4 LTS
- Windows 2003
- Windows 2003 R2
- Windows 2008
- Windows XP
- Windows Vista
- Solaris
- HP



The following architecture is no longer supported:

- POWER4 processor-based systems (DB2 servers and data clients)
- See IBM Knowledge Center for details on supported Operating Systems.

Starting in DB2 11.1, some operating systems, Linux distributions and architecture are no longer supported.

When upgrading OS what is required for DB2

1. Ensure have good offline backup of databases
2. Stop DB2 completely before OS upgrade:
db2stop
db2admin stop
3. Do patch upgrade (keep track of what patches are so know differences)
4. Ensure kernel settings are still the same after patch.
5. Restart DB2:
db2start
db2admin start
6. Run sanity check against database ensuring that can connect to database and select from tables.
7. If there is change to /usr/lib or /usr/lib64, should run **db2iupdt** against each of the instances.

Supported Client Access and Client Upgrade

- **ALWAYS UPGRADE SERVERS** before upgrading **CLIENTS!**
- **DB2 11.1, 10.5, 10.1, and 9.7 clients and drivers can access remote DB2 11.1 Server:**
 - IBM Data Server Driver Package
 - IBM Data Server Driver for JDBC and SQLJ
 - IBM Data Server Driver for ODBC and CLI
 - IBM Data Server Runtime Client
 - IBM Data Server Client
- **If use pre-11.1 clients to access DB2 11.1 servers, applications are restricted to functionality available for that client.**
- If different versions of client and DB2 server located on same system, local client-to-server connections using IPC are not supported. Establish connection as remote connection (loopback connection) using TCP/IP.
- When later-level client accesses earlier-level server, later level client functionality not available to server.
- Upgrading client involves installing DB2 11.1 client copy and then upgrading client instance.
- Current level of client installed determines way to proceed with upgrade to DB2 11.1.
 - **Can directly upgrade to DB2 11.1 clients from DB2 10.5, DB2 10.1, or DB2 9.7.**
 - If have DB2 9.5 or earlier clients, upgrade to any DB2 10.1 or DB2 9.7 client first.
- Which JDBC driver (JCC) version corresponds with each DB2 release and Fix Pack level?
 - <http://www-01.ibm.com/support/docview.wss?uid=swg21363866>
 - For JDBC or SQLJ applications, if you are using the IBM DB2 Driver for SQLJ and JDBC, you can determine the level of the driver by running `db2jcc -version`



UPGRADE

14

© 2016 IBM Corporation

A client instance allows connecting application to database and keeps information about client configuration, cataloged nodes, and cataloged databases. The upgrade process consists of all the tasks that you must perform to have your DB2 environment running successfully after upgrading. These components are DB2 servers, DB2 clients, database applications, and routines. Upgrading DB2 servers to DB2 11.1 requires an understanding of upgrade concepts, upgrade restrictions, upgrade recommendations, and your DB2 server. When you have a complete understanding of what upgrading your DB2 server involves, you can create your own upgrade plan. The upgrade of each of the components in your DB2 environment requires that you perform different tasks:

- *Upgrading DB2 servers* involves upgrading your existing instances and databases so that they can run in the new release.
- *Upgrading clients* involves upgrading your client instances to keep the configuration of your existing clients.
- *Upgrading database applications and routines* involves testing them in the new release and modifying them only when you must support changes in this new release.

The IBM clients and drivers are as follows:

- IBM Data Server Driver Package
- IBM Data Server Driver for JDBC and SQLJ
- IBM Data Server Driver for ODBC and CLI
- IBM Data Server Runtime Client
- IBM Data Server Client

Product packaging enhancements for DB2 11.1 (1 of 2)

- **DB2 Advanced Enterprise Server Edition**
 - No processor, memory, or database size limits.
 - Includes all functions found in DB2 Enterprise Server Edition plus column organized tables, in-memory database, data compression, workload management, replication, and distributed partitioning capability.
 - Includes full complement of warehouse tools and Data Server Manager Enterprise. The included tools must be installed separately.
 - Functionality provided by DB2 Performance Management Offering.
- **DB2 Advanced Workgroup Server Edition**
 - Similar to DB2 Advanced ESE, except places limits on processor and memory.
- **DB2 Enterprise Server Edition**
 - No processor, memory, or database size limits.
 - Includes all functions found in DB2 WSE plus materialized query tables.
 - includes Data Server Manager Base which requires separate installation.
 - DB2 Performance Management Offering can be added by activating license.



There are multiple DB2 database product editions, each with a unique combination of functionality and DB2 add-on offerings. For details of entitlement, rights and obligations, refer to the license agreement for your DB2 product or offering.

For details of functionality included in each DB2 database product edition, see the DB2 LUW 11.1 Knowledge Center.

Product packaging enhancements for DB2 11.1 (2 of 2)

- **DB2 Workgroup Server Edition**
 - Places limits on processor and memory.
 - Includes Data Server Manager Base, which requires separate installation.
 - DB2 Performance Management Offering can be added by activating license.
- **DB2 Express-C**
 - Free, entry-level edition of DB2 data server for developer and partner community.
 - Includes self-management features and embodies all of core capabilities.
 - Can be used for development and deployment at no charge and can be distributed with third-party solutions without any royalties to IBM.
 - Refreshed at major release milestones. Comes with online community-based assistance.
 - No DB2 add-on offerings can be added.
 - **There is no DB2 Express Server Edition in DB2 11.1.**
- **DB2 Developer Edition**
 - Package for single application developer to design, build, and prototype applications for deployment on any of client or server platforms.
 - Includes all DB2 server editions, DB2 Connect Enterprise Edition, allowing you to build solutions that use latest data server technologies.
 - Cannot be used for production systems. Must acquire separate user license for each Authorized User of this product.
 - DB2 Performance Management Offering included.



There are multiple DB2 database product editions, each with a unique combination of functionality and DB2 add-on offerings. For details of entitlement, rights and obligations, refer to the license agreement for your DB2 product or offering.

For details of functionality included in each DB2 database product edition, see the DB2 LUW 11.1 Knowledge Center.

Product packaging: DB2 11.1 add-on offerings

- **IBM DB2 Performance Management Offering**
 - Provides IBM Data Server Manager and Workload Manager functionality.
 - Can be added to DB2 Enterprise Server Edition or DB2 Workgroup Server Edition.
 - Has its own license certificate file.
- **IBM DB2 Advanced Recovery Feature**
 - Offers additional tools to protect mission-critical data
 - DB2 Recovery Expert
 - DB2 Merge Backup
 - IBM InfoSphere Optim High Performance Unload
- **Add-on offerings now included in some DB2 11.1 product editions:**
 - IBM DB2 BLU Acceleration In-Memory Offering
 - IBM DB2 Business Application Continuity Offering
 - IBM DB2 Encryption Offering
- **Update some DB2 editions to another one by just updating license file.**
 - Do not have to reinstall DB2 database product.
 - Can also update some DB2 editions to enable DB2 Connect functionality by adding your DB2 Connect license file.



In DB2 Version 11.1, you can add functionality to various DB2 base editions.

For details of functionality included in each DB2 database product edition, see the DB2 LUW 11.1 Knowledge Center.

Functionality in DB2 database product editions (1 of 5)

	EXPRESS-C	Workgroup Server	Enterprise Server	Advanced Workgroup Server	Advanced Enterprise Server	Developer Edition
Access Plan Reuse	No	Yes	Yes	Yes	Yes	Yes
Adaptive Compression & Classic Row Compression	No	No	No	Yes	Yes	Yes
Compression: Backup	Yes	Yes	Yes	Yes	Yes	Yes
Connection Concentrator	No	Yes	Yes	Yes	Yes	Yes
Continuous Data Ingest	Yes	Yes	Yes	Yes	Yes	Yes
Data Server Manager Base	No	Yes	Yes	No	No	No
Data Server Manager Enterprise	No	IBM DB2 Performance Management Offering	IBM DB2 Performance Management Offering	Yes	Yes	Yes
DB2 Advanced Copy Services	No	Yes	Yes	Yes	Yes	Yes
DB2 column-organized tables-5	No	No	No	Yes	Yes	Yes
DB2 Connect functionality	No	No	No	Yes - 1	Yes - 1	Yes

Some functionality is available in only certain DB2 database product editions.

The table indicates which functionality is included in a DB2 product edition and DB2 offering.

As IBM data servers continue to evolve, the names and the packaging of the DB2 components change to respond to market needs.

IBM has updated product packaging to provide simplicity through fewer packages, and greater value through more function and features included in the base DB2 editions.

Functionality in DB2 database product editions (2 of 5)

	EXPRESS-C	Workgroup Server	Enterprise Server	Advanced Workgroup Server	Advanced Enterprise Server	Developer Edition
DB2 pureScale	No	Yes	Yes	Yes	Yes	Yes
Encryption - native backup encryption	Yes	Yes	Yes	Yes	Yes	Yes
Encryption - native database encryption	Yes	Yes	Yes	Yes	Yes	Yes
Federation with IBM data servers	Yes	Yes	Yes	Yes	Yes	Yes
Federation with DB2 for I and DB2 z data sources	No	No	No	Yes - 8	Yes - 8	Yes
Federation with all other data sources including Oracle	No	No	No	Yes	Yes	Yes
HADR	No	Yes	Yes	Yes	Yes	Yes
IBM Data Studio	Yes	Yes	Yes	Yes	Yes	Yes
IBM(r) DB2 Merge Backup	No	IBM DB2 Advanced Recovery Feature				

1. This functionality is availability only by using the SQL Warehouse tool (SQW).
2. DB2 Advanced Enterprise Server Edition includes 10 InfoSphere Data Architect user licenses.
3. Replication tools except the Replication Center are available on all supported operating systems. The Replication Center is available only on Linux and Windows operating systems.
4. IBM InfoSphere Change Data Capture software components are included for support of:
 - Unidirectional replication without data transformations between a single DB2 for LUW source and no more than two DB2 for LUW servers of the same version for the purpose of HADR
 - For supporting shadow table functionality within a single DB2 for LUW instance to replicate from row-organized tables into column-organized tables.

Functionality in DB2 database product editions (3 of 5)

	EXPRESS-C	Workgroup Server	Enterprise Server	Advanced Workgroup Server	Advanced Enterprise Server	Developer Edition
IBM InfoSphere Change Data Capture for DB2 for LUW - 4	No	No	No	Yes	Yes	Yes
IBM InfoSphere CDC Access Server - 4	No	No	No	Yes	Yes	Yes
IBM InfoSphere CDC Management Console - 4	No	No	No	Yes	Yes	Yes
IBM InfoSphere Data Architect	No	No	No	Yes - 2	Yes - 2	Yes - 2
IBM InfoSphere Optim High Performance Unload	No	IBM DB2 Advanced Recovery Feature				
Index Compression	No	No	No	Yes	Yes	Yes
Label-based access control (LBAC)	No	Yes	Yes	Yes	Yes	Yes
Materialized Query Tables	No	No	Yes	Yes	Yes	Yes
Multidimensional Clustering tables	No	Yes	Yes	Yes	Yes	Yes

20

© 2016 IBM Corporation

- Includes support for:
 - Column-organized tables,
 - Shadow tables (a column-organized copy of a row-organized table), and
 - InfoSphere Change Data Capture (for replication of the shadow table).
- In DB2 10.5 Fix Pack 5 and later fix packs, this functionality is available with the DB2 offering listed. In addition to applying the base license certificate file for the DB2 database product edition, also apply the license certificate file for the offering. For details about the functionality available with each DB2 offering, see [DB2 database product editions and DB2 offerings](#). For the license file names, see [DB2 license files](#).
- The support for the DB2 Workgroup Server Edition and the DB2 Enterprise Server Edition on pureScale is only for one primary server and one standby server. The standby server can only be used to perform administrative tasks, or be ready to take over from the primary server in the event of a planned or unplanned outage.
- Federation with DB2 for i and DB2 for z data sources is possible with sufficient entitlement to DB2 Connect.

Functionality in DB2 database product editions (4 of 5)

	EXPRESS-C	Workgroup Server	Enterprise Server	Advanced Workgroup Server	Advanced Enterprise Server	Developer Edition
Multi-Temperature Storage	No	Yes	Yes	Yes	Yes	Yes
Net Search Extender	Yes	Yes	Yes	Yes	Yes	Yes
Online Reorganization	No	Yes	Yes	Yes	Yes	Yes
Oracle Compatibility	Yes	Yes	Yes	Yes	Yes	Yes
Partitioning - partitioned database environment	No	No	No	Yes	Yes	Yes
Partitioning - Table partitioning	No	Yes	Yes	Yes	Yes	Yes
pureXML storage	Yes	Yes	Yes	Yes	Yes	Yes
Q Replication with two other DB2 LUW servers	No	No	No	Yes	Yes	Yes
Query Parallelism	No	Yes	Yes	Yes	Yes	Yes
Replication Tools	Yes - 3	Yes - 3	Yes - 3	Yes	Yes - 3	Yes
Row & Column Access Control (RCAC)	No	Yes	Yes	Yes	Yes	Yes

Some functionality is available in only certain DB2 database product editions.

Functionality in DB2 database product editions (5 of 5)

	EXPRESS-C	Workgroup Server	Enterprise Server	Advanced Workgroup Server	Advanced Enterprise Server	Developer Edition
Scan Sharing	No	Yes	Yes	Yes	Yes	Yes
Spatial Extender	Yes	Yes	Yes	Yes	Yes	Yes
SQL Replication with DB2 LUW databases	No	Yes	Yes	Yes	Yes	Yes
Time Travel Query	Yes	Yes	Yes	Yes	Yes	Yes
IBM Tivoli System Automation for Multiplatforms	No	Yes	Yes	Yes	Yes	Yes
Workload management	No	IBM DB2 Performance Management Offering - 6	IBM DB2 Performance Management Offering - 6	Yes	Yes	Yes

This table (pages 1 through 5) is for informational purposes only. For details of entitlement, rights and obligations, refer to the license agreement for your DB2 product and offering.

New DB2 Direct Editions

- **New Delivery Mechanisms for DB2 licenses**
 - New license metrics to facilitate hybrid cloud deployments
 - Option to deploy either on-premises or on cloud
- **Two Versions depending on requirements**
 - **DB2 Direct Standard Edition 11.1**
 - Has all database features of DB2 WSE
 - **DB2 Direct Advanced Edition 11.1**
 - Has all database features of DB2 Advanced ESE
 - Both include access to previous (10.5) editions of DB2
- **New Simplified Licensing Metric**
 - Virtual Processor Core (VPC) sold as monthly license charge
 - 1 VPC for every VPC available to a virtual Servers Operating System, VPC for each physical core of non-partitioned physical server
- **Predictable Maintenance windows**
 - Sliding 24 month version N, N-1 Support



The DB2 Direct Advanced Edition has all DB2 Server and Client features from DB2 Advanced Server Edition including encryption, multitenant deployments, adaptive compression, BLU Acceleration, SQL compatibility with PL/SQL, Data Server Manager, pureScale and database partitioning feature options. It also includes federation capabilities providing access to non-DB2 database sources like Oracle, MS SQL, Teradata, Hadoop, Netezza, Spark and other solutions. It also includes access to 10 User licenses of Infosphere Data Architect per installation for designing and deploying database implementations.

DB2 Direct Standard Edition is modelled on DB2 Workgroup Edition, which provides encryption, pureScale for Continuously available HA deployments, Multitenant Deployments, SQL compatibility with PL/SQL, Data Server Manager Base Edition, Table partitioning, multi-dimensional clustering, parallel query and concurrent Connection pooling. It is limited to 16 cores and 128GB of RAM and is ideal for small to mid-sized database applications providing enterprise level availability, Query performance and Security as well as unlimited database size

You can take advantage of the new subscription model to lower costs and enjoy licensing flexibility for on-premises and cloud deployments.

Virtual ProcessorCore (VPC) Charge metric - Virtual processor core licensing gives you flexibility and simplified sub capacity licensing options that enables you to optimize your licensing to meet your business requirements. There are two Licensing Scenarios you can apply:

1. Simply license the sum of all available Virtual Processor Cores on all Virtual Servers the Direct edition is installed on OR
2. when you can identify a Server and it is more cost effective to do so simply license all available Processor Cores on the Physical Server regardless of the number of virtual machines on the system.

Benefits: This makes it simple for private and public Cloud deployments alike and enables you to optimize your licensing.

sub-capacity situations. A Virtual Processor Core is a Processor Core in an unpartitioned Physical Server, or a virtual core assigned to a Virtual Server. The Licensee must obtain entitlement for each Virtual Processor Core made available to the Program.

Plan your DB2 environment upgrade

1. Review new, deprecated, discontinued functionality for DB2 11.1 and for any releases between release are upgrading from and DB2 11.1.
2. Plan how to modify database applications and routines. Ensure they run successfully in DB2 11.1.
3. Set up DB2 11.1 test server and create test databases.
4. Test database applications and routines on DB2 11.1 test database to determine whether run successfully. If application requires a client, use DB2 11.1 client.
5. Upgrade DB2 servers and clients in test environment. Determine what issues are and how to resolve. Use info to adjust upgrade plan.
6. Upgrade DB2 servers to DB2 11.1 in production environment. Ensure they operate as expected.
7. Upgrade clients to DB2 11.1 in production environment. Ensure clients operate as expected.
8. Test database applications and routines in DB2 11.1 upgraded production environment to determine whether run as expected.
9. Make upgraded environment available to users.
10. Plan to remove use of deprecated functionality and incorporate new functionality from DB2 11.1.



24

© 2016 IBM Corporation

Your environment has several components such as DB2 servers, DB2 clients, database applications, scripts, routines and tools. Planning your upgrade requires a thorough understanding of the upgrade process of each component in your environment.

First, devise a strategy on how to approach your environment upgrade. You must determine the order in which you are going to upgrade each component. The characteristics of your environment and the information in upgrade essentials, especially the best practices and restrictions, can help you determine your strategy.

The above list is an example of a good *upgrade strategy* in which you test your database applications and routines and determine that they run successfully in DB2 10.5.

After you have a strategy that will give you the outline for your upgrade plan, you can define the upgrade plan details for each component in your environment. An *upgrade plan* should include for each component:

- Upgrade prerequisites
- Pre-upgrade tasks
- Upgrade tasks
- Post-upgrade tasks

If you have previous upgrade plans, review them and compare them with the upgrade plan for DB2 10.5. Include in your new plan any steps related to internal procedures to request access, software installation or other system services within your organization.

Finally, plan to remove the use of deprecated functionality and incorporate new functionality from DB2 10.5. Although you are only required to remove the use of discontinued functionality, you should also plan to remove the use of deprecated functionality after upgrade because they will become unsupported in a future release. Also, you should take advantage of new functionality for your database products, applications, and routines to enhance functionality and improve performance.

Benchmark DB2 server performance – before and after

- Run performance tests before upgrading your DB2 server.
- Use **db2batch** benchmark tool to collect elapsed and CPU times for running queries.

```
db2batch -d DBX -f input.sql -r output.fil -i complete -o e yes -isol CS
```

- Record exact environment conditions where run your tests.
- Keep record of **db2exfmt** command output for each test query.
- Compare the results before and after upgrade.
- Identify and correct any performance degradation that occurs.



Run a number of performance tests before upgrading your DB2 server. The db2batch benchmark tool helps you to collect elapsed and CPU times for running queries. You can use this tool to develop performance tests. Record the exact environment conditions where you run your tests.

Also, keep a record of the db2exfmt command output for each test query. Compare the results before and after upgrade. This practice can help to identify and correct any performance degradation that might occur.

Download DB2 LUW Code

▪ **To download purchased product, go to [IBM Passport Advantage Web site](#) and select products using one of these options:**

1. Find by search text using product names as shown below
2. Find by part number if know Assembly Part Number (see your order)
3. Find by categories:
 - Select Brand = "Information Management"
 - Select Family name ="IBM® Database 2"
 - Select Platform = for example, AIX®, Windows®, Linux®

▪ **Required and optional parts for each product are listed, and described in detail.**

- There is option to download each licensed eAssembly.
- Each eAssembly contains all of required and optional downloads needed for particular product and platform combination.

To download a trial version of various DB23 database products, go to <http://www.ibm.com/software/data/db2/linux-unix-windows/download.html>

To download DB2 recommended fix packs, go to: <http://www.ibm.com/support/docview.wss?uid=swg21321001>

db2prereqcheck command

- Check if system meets DB2 11.1.0 prerequisites:
 - **db2prereqcheck –v 11.1.0.0 -s**

-v <version>

-s Print prerequisite validation summary on screen

-c Checks prerequisites for thin client

-p Checks prerequisites for DB2 pureScale environment

-i Checks the non-DB2 pureScale prerequisites for the latest DB2 version that is defined in the resource XML file.

If do not specify any options, both DB2 pureScale and non-DB2 pureScale server prerequisites checked for all DB2 versions listed in DB2prereqs.xml.

The -i and -v parameters are mutually exclusive.

-o <report-file>

- Uses resource XML file that contains prerequisites. Default path of XML file located in: <DB2 installation>/cfg/DB2prereqs.xml



27

© 2016 IBM Corporation

The supported UNIX, Linux and Windows operating systems have changed in DB2 11.1. Review the installation requirements for DB2 servers and IBM data server clients to determine whether your operating system version is supported and if you need to upgrade your operating system before installing DB2 11.1. Newer versions of operating systems can also bring new hardware requirements. Performing hardware and operating system upgrades separately from DB2 database product upgrade simplifies problem determination if you encounter upgrade difficulties. If you upgrade your software or hardware before a DB2 database product upgrade, ensure that your system is operating as expected before attempting to upgrade your DB2 database product. Checks whether your system meets the prerequisites for the installation of a specific version of DB2 Database for Linux, UNIX, and Windows. By using the db2prereqcheck command, you can determine whether your system satisfies the prerequisites before you download DB2 Database for Linux, UNIX, and Windows and start the installation process. The db2prereqcheck command uses a resource XML file that contains the prerequisites. The default path of the XML file is located in DB2 installation/cfg/DB2prereqs.xml. You must have read or write permissions on the XML file. Do not modify the contents of the XML file.

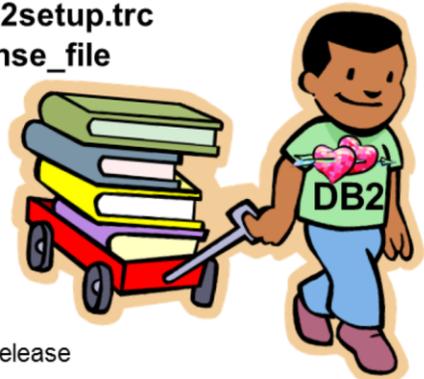
- -c Checks the prerequisites for thin client. The -c, -u, -g, and -p parameters are mutually exclusive.
- -u Checks the uDAPL requirement for the DB2 pureScale environment (Linux and AIX only). The -c, -u, -g, and -p parameters are mutually exclusive. On RoCE networks, you must ensure the AIX and uDAPL software prerequisites are satisfied. The db2prereqcheck command does not automatically validate these levels.
- -g Checks the GPL compilation requirement (Linux only). The -c, -u, -g, and -p parameters are mutually exclusive.
- -p Checks the prerequisites for the DB2 pureScale environment (Linux and AIX only). The -c, -u, -g, and -p parameters are mutually exclusive.
- -t network configuration type. The -t parameter validates prerequisites requirement for a specific type of network configuration (Linux only). The network configuration type must be SINGLE_IB_PORT_CLUSTER, MULT_IB_PORT_CLUSTER, SINGLE_ROCE_PORT_CLUSTER or MULTI_ROCE_PORT_CLUSTER.
- -s Print prerequisite validation summary on screen.
- -i Checks the prerequisites for the latest DB2 version that is defined in the resource XML file.
- To display the latest version number in the XML file, run -i with -s. The -i and -v parameters are mutually exclusive.
- -v version - Checks the prerequisites for the specific DB2 version. You can use this parameter with other db2prereqcheck command parameters. The -i and -v parameters are mutually exclusive.
- -l - Checks the prerequisites related to pureScale and unrelated to DB2 pureScale environments for the latest DB2 version that is defined in the XML resource file. The -l and -v parameters are mutually exclusive.
- -f xml-file - Specifies the name of the XML resource file. The default XML file (with file path : DB2 installation/cfg/DB2prereqs.xml) will be used if the -f parameter is not specified.
- -o report-file - Specifies a name for the report file. For example, db2prereqcheck.rpt.
- hl list_of_hostnames - Specifies a list of hosts that are checked for passwordless root SSH access between all hosts.

Reference: Student Notes

- . -dl *list_of_shared_devices* - Specifies one or more PVIDs (on AIX operating systems) or WWNs (on Linux operating systems) of the shared disks that are verified to make sure that they are accessible by all hosts. If you specify this parameter, you must also specify the -hl parameter.
- . -iu *instance_user* - Specifies the instance user name. The UID and GID of the instance user name are verified to make sure they are the same across all hosts. If you specify this parameter, you must also specify the -hl parameter.
- . -nl *list_of_netnames* - Specifies the list of netnames. These netnames are used to ping the RocE & IB networks to verify that they are pingable between all hosts.
- . -upgrade - Checks whether the required entry in the netmon.cf file is correct.

DB2 INSTALLATION METHODS

- On Linux/UNIX, use **db2setup** command or **response file** installation method.
 - Located on DB2 database installation media. Launches DB2 Setup wizard to define installation and install DB2 database products.
 - **./db2setup -l /temp/db2setup.log -t /tmp/db2setup.trc**
 - **./db2setup -r responsefile_directory/response_file**
 - -r option performs installation without further input, taking installation configuration information from response file.
 - -c validates contents of response file before install fix pack. Ensures contents of response file correct before installation.
- On Windows, use **setup** command or **response file** installation method.
 - **setup -l c:\temp\db2setup.log -t c:\tmp\db2setup.trc**
 - **setup -u c:\ responsefile_directory/response_file**



- **db2_install** is deprecated and might be removed in future release

29

© 2016 IBM Corporation

The `db2setup` command queries the operating system to determine the existing language settings. If the language setting of your operating system is supported by `db2setup`, then that language will be used when displaying the DB2 Setup wizard. If your system uses the same code pages but different locale names than those supported by the DB2 interface, you can still see the translated `db2setup` by setting your `LANG` environment variable to the appropriate value. The `db2setup` command is only available on Linux and UNIX operating systems. The command for Windows operating systems is **setup**.

To start the IBM DB2 Setup Launchpad, run the `db2setup` command from the directory where the DB2 installation image resides:

```
./db2setup -l /temp/db2setup.log -t /tmp/db2setup.trc
```

- The `-l` and `-t` recommended parameters enable DB2 logging and tracing.

To create a response file use the DB2 Setup wizard:

- Type the `db2setup` command to launch the DB2 Instance Setup Wizard.
- On the Select installation, response file creation or both panels, select the **Save my installation settings in a response file** option.

After creating a response file, you can make changes to the response file to activate or deactivate keywords.

```
./db2setup -r responsefile_directory/response_file
```

DB2 Setup wizard: Installing Windows servers



1. Log on to system with local Administrator account defined for DB2 installation.
2. Stop all DB2 instances, services and applications.
3. If have DB2 database product DVD, insert it into drive. If enabled, autorun feature automatically starts DB2 Setup Launchpad. If autorun does not work, use Windows Explorer to browse DB2 database product DVD and double-click setup icon to start DB2 Setup Launchpad. Can also run **setup.exe** to launch DB2 wizard.
4. If downloaded DB2 database product from Passport Advantage, run executable file to extract DB2 database product installation files. Use Windows Explorer to browse DB2 installation files and double-click setup icon to start DB2 Setup Launchpad.
5. From DB2 Setup Launchpad, view installation prerequisites and release notes, or proceed directly to installation with following choices:
 - To update existing DB2 copy and update all instances running on this DB2 copy, select Work with Existing in Install a Product panel. Then select DB2 copy want to update with update action. You cannot install additional functionality with this action.
 - To install new DB2 copy and selectively update instances running on existing DB2 copy to new copy after installation, select Install New in Install a Product panel. Allows you to also install additional functionality.
 - To add functionality to existing DB2 copy, select Work with Existing in Install a Product panel. Then select DB2 copy that want to update with Add new function action. Action only available when DB2 copy is at same release level as install image.
6. DB2 Setup wizard determines system language, and launches setup program for that language. Online help is available to guide you through remaining steps. To invoke online help, click Help or press F1. Can click Cancel at any time to end installation.
7. Sample panels when using DB2 setup wizard lead you to installation process.
8. IBM Data Studio can be installed by running the DB2 Setup wizard.
9. You will not be able to update more than one DB2 copy at the same time. In order to update other DB2 copies that might be installed on the same computer, you must rerun the installation.
10. Once installed or updated a DB2 copy, can always update instances that run in other DB2 copies of same release, to run on this new DB2 copy by issuing **db2iupdt** command.

30

© 2016 IBM Corporation

Your DB2 database product is installed, by default, in the *Program_Files\IBM\sqllib* directory, where *Program_Files* represents the location of the Program Files directory.

If you are installing on a system where this directory is already being used, the DB2 database product installation path has *_xx* added to it, where *xx* are digits, starting at 01 and increasing depending on how many DB2 copies you have installed.

You can also specify your own DB2 database product installation path.

For information about errors encountered during installation, review the installation log file located in the My Documents\DB2LOG\ directory. The log file uses the following format: *DB2-ProductAbrev-DateTime.log*.

If you want your DB2 database product to have access to DB2 documentation either on your local computer or on another computer on your network, then you must install the *DB2 Knowledge Center*. The *DB2 Knowledge Center* contains documentation for the DB2 database system and DB2 related products. By default, DB2 information is accessed from the web if the *DB2 Knowledge Center* is not locally installed.

Windows Installation Considerations

- During installation on Windows, if select **DB2 copy with upgrade action** in **Work with Existing** window and have local databases cataloged on instances, a message box warns that you must run **db2ckupgrade** from DB2 database product CD.
- Then can choose one of following actions:
 - Ignore message and continue the installation process.
 - Run **db2ckupgrade**. If command runs successfully, continue installation process. If find errors, quit installation process, fix any errors, and then rerun installation process.
 - Quit installation process.



Before you upgrade your DB2 server, review the upgrade essentials for DB2 servers, including recommendations, restrictions, and disk space requirements to identify the changes or restrictions that can affect your upgrade. You must be ready to address any issues before upgrade in order to have a successful upgrade.

How to verify fix pack level for Windows installation image?

1) CD to installation image location where DB2 installation file was uncompressed.

2) Locate directory called ...**db2\Windows** or ...**image\db2\Windows**



3) Look for any *.**prd** file since they will all be at same fixpack level. Examples:

AESE.**prd** - DB2 Advanced Server Edition
CONSV.**prd** - DB2 Connect Server Edition
ESE.**prd** - DB2 Enterprise Server Edition
WSE.**prd** - DB2 Workgroup Server Edition

4) File *.**prd** contains fixpack number.

Modification=400 refers to fixpack 4 so installation image is for DB2 v10.1 Fixpack 4.

VERSION=10, RELEASE=1, **MODIFICATION=400**, SERVICE_LEVEL=IP23579

The above steps can be used to verify the fix pack level for the installation media used to install DB2 on Windows.

DB2 Setup wizard: Installing Linux/UNIX servers

- If have physical DB2 database product DVD, change to directory where DB2 database product DVD is mounted by entering: `cd /dvdrom` where */dvdrom* represents mount point of DB2 database product DVD.
- If downloaded DB2 database product image, extract product file.
 - `gzip -d product.tar.gz` where *product* name of the downloaded product.
- Untar on Linux:
 - `tar -xvf product.tar`
- Untar on AIX, HP-UX, and Solaris:
 - `guntar -xvf product.tar` where *product* name of the downloaded product.
- Change directory: `cd ./product` where *product* name of the downloaded product.
- **Note:** If downloaded National Language Package, untar into same directory which creates subdirectories (for example *./nlpack*) in same directory, and allows installer to automatically find installation images without prompting.
- DB2 Setup wizard is graphical installer. Must have X windows software capable of rendering graphical user interface. Export your display: `export DISPLAY=9.26.163.144:0`
- Enter `./db2setup` from directory where database product image resides to start DB2 Setup wizard.
- IBM DB2 Setup Launchpad opens. From window, can view installation prerequisites and release notes, or can proceed directly to installation. Can also review installation prerequisites and release notes for late-breaking information.
- Click Install a Product and Install a Product window will display products available for installation.
- Launch installation by clicking Install New. Proceed through installation following DB2 Setup wizard's prompts.
- Sample panels when using DB2 setup wizard will lead you to installation process.
- After have initiated installation, proceed through DB2 Setup wizard installation panels and make selections. Installation help is available to guide through remaining steps. To invoke installation help, click Help or press F1. Can click Cancel at any time to end installation.



33

© 2016 IBM Corporation

For non-root installations, DB2 database products are always installed in the *\$HOME*/sqlib directory, where *\$HOME* represents the non-root user's home directory. For root installations, DB2 database products are installed, by default, in one of the following directories:

- **AIX, HP-UX, and Solaris:** `/opt/IBM/db2/11.1`
- **Linux:** `/opt/ibm/db2/11.1`

If you are installing on a system where this directory is already being used, the DB2 database product installation path will have `_xx` added to it, where `_xx` are digits, starting at 01 and increasing depending on how many DB2 copies you have installed.

You can also specify your own DB2 database product installation path.

DB2 installation paths have the following rules:

- Can include lowercase letters (a–z), uppercase letters (A–Z), and the underscore character (_)
- Cannot exceed 128 characters
- Cannot contain spaces
- Cannot contain non-English characters

The installation log files are:

- The DB2 setup log file. This file captures all DB2 installation information including errors.
 - For root installations, the DB2 setup log file name is `db2setup.log`.
 - For non-root installations, the DB2 setup log file name is `db2setup_username.log`, where *username* is the non-root user ID under which the installation was performed.
- The DB2 error log file. This file captures any error output that is returned by Java (for example, exceptions and trap information).
 - For root installations, the DB2 error log file name is `db2setup.err`.
 - For non-root installations, the DB2 error log file name is `db2setup_username.err`, where *username* is the non-root user ID under which the installation was performed.

By default, these log files are located in the `/tmp` directory. You can specify the location of the log files.

There is no longer a `db2setup.his` file. Instead, the DB2 installer saves a copy of the DB2 setup log file in the `DB2_DIR/install/logs/` directory, and renames it `db2install.history`. If the name already exists, then the DB2 installer renames it `db2install.history.xxxx`, where `xxxx` is 0000-9999, depending on the number of installations you have on that machine.

Student Notes: Reference Only

Each installation copy has a separate list of history files. If an installation copy is removed, the history files under this install path will be removed as well. This copying action is done near the end of the installation and if the program is stopped or aborted before completion, then the history file will not be created.

Verify installation using CLP – SAMPLE Database

1. Create new 11.1 instance.
#DB2DIR/instance/db2icrt -u db2fenc1 db2inst1
2. Log on to system as user with SYSADM authority.
3. Start database manager
.db2start
4. Create the SAMPLE database.
.db2sampl
 - .Can take a few minutes to process. No completion message; when command prompt returns, process is complete.
 - .SAMPLE database is automatically cataloged with database alias SAMPLE when created.
 - .On non-partitioned created with XML objects. On partitioned database can use **-xml** option to create XML objects.
5. Connect to SAMPLE database, retrieve a list of employees that work in department 20, and reset the database connection.
 - .db2 connect to sample**
 - .db2 "select * from staff where dept = 20"**
 - .db2 connect reset**
6. After verified installation, remove SAMPLE database to free disk space.
 - .db2 drop database sample**
 - ./opt/IBM/db2/copy1/instance/db2idrop db2inst1**



35

© 2016 IBM Corporation

You can verify the installation by creating the SAMPLE database and running SQL commands to retrieve sample data. The SAMPLE database component, found in the features selection, must be installed on your system and is included in a typical installation. You require a user with SYSADM authority.

In non-partitioned database environments:

- Creates a database named SAMPLE with a Unicode (UTF-8) code set in the default database path.
- Creates relational database objects including tables, indexes, constraints, triggers, functions, procedures, multi-dimensional clustered tables and materialized query tables.
- Populates relational tables with data.
- Creates tables with XML data type columns.
- Creates indexes over XML data.
- Creates an XML schema repository that contains XML schema documents.
- All database object names are qualified with the value of the CURRENT_SCHEMA special register.

Student Notes: Reference Only

In partitioned database environments:

- Creates a database named SAMPLE with a Unicode (UTF-8) code set in the default database path.
- Creates relational database objects including tables, indexes, constraints, triggers, functions, procedures, multi-dimensional clustered tables and materialized query tables.
- Populates tables with data.
- All database object names are qualified with the value of the CURRENT_SCHEMA special register.
- The –xml option can be used to create tables with columns of data type XML, creates indexes on the XML columns, registers XML schemas, and populates these tables with data including XML document values. This option is only supported where XML is supported. If XML is not supported, this option is ignored.

Checklist: First steps after installation

1. [Deleting a Firefox browser profile](#)

After run DB2 Setup wizard or DB2 First Steps, may want to delete browser profile to prevent JavaScript from running automatically when visiting other Web pages with this profile.

2. [Applying DB2 licenses](#)

To license your product, register appropriate license key with DB2 database product.

3. [Updating DB2 licenses](#)

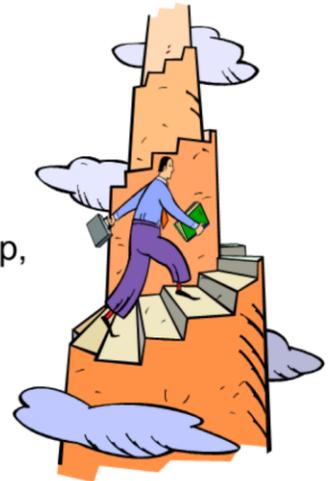
Can switch between editions in DB2 11.1 - Workgroup, Enterprise, and Advanced Editions common image.

4. [Post-installation tasks for DB2 database servers](#)

After installing DB2 database, you must complete a number of tasks.

5. OPTIONAL: [First Steps interface](#)

Launches First Steps interface, which contains links to functions that you need to begin learning about and using DB2 product.



After you install a DB2 database product, there are additional necessary and optional tasks available.

1. Deleting Firefox browser profile:

- If any Firefox browsers are open, close them.
- Open the Firefox Profile Manager.
 - On Windows
 - Open the Windows Start menu and select Run...
 - Type `firefox.exe -profilemanager` and click OK. The Firefox - Choose User Profile dialog opens.
 - On Linux and UNIX
 - Open a command prompt.
 - Switch to the Firefox directory.
 - Start Firefox with the `-profilemanager` switch. For example: `./firefox -profilemanager`
- Select the `DB2_FIRSTSTEPS` profile.
- Click Delete Profile...
- Click Exit to close the Firefox - Choose User Profile dialog.



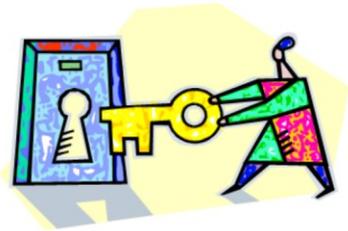
If you use this profile to view other Web pages that have JavaScript, the JavaScript will run automatically without warnings.

If you use Firefox as your Web browser, a browser profile is created when you run DB2 First Steps. If you agree, a browser profile named `DB2_FIRSTSTEPS` is created.

2. Applying DB2 licenses – db2licm

- If want license key added automatically during product installation, copy license key to **/db2/license** directory of installation image before launching DB2 Setup wizard.
- If do not add any license files to /db2/license directory, see *"License not registered"* message from **db2licm -l** command.

- Can apply licenses manually after installation by running **db2licm -a**.



- If register valid license key, **db2licm -l** command successfully lists all products with available license information.

To license your product, register the appropriate license key with your DB2 database product.

If you want the license key added automatically during product installation, copy the license key to the /db2/license directory of the installation image before launching the DB2 Setup wizard.

If you do not add any license files to the /db2/license directory, you will see a "License not registered" message in the output from the db2licm -l command. The following base licenses are available at the /db2/license directory of the server image and the product-specific base license is automatically applied during DB2 product installation.

- db2ese.lic
- db2exp.lic
- db2consv.lic

You can apply the licenses manually after the installation by running the db2licm -a command. If you register a valid license key, the db2licm -l command will successfully list all the products with available license information

To use DB2 features or to be in compliance with your current product entitlement, you might be required to register additional license keys.

How to download DB2 license?

- **License files for DB2 11.1 are shipped separately.**
 - Done for convenience so downloading license file takes less time due to small size.
 - Download license activation key from Passport Advantage.
 - Then install it.
- How do I download my DB2 license from Passport Advantage?
 - <http://www-01.ibm.com/support/docview.wss?uid=swg21305417>
- DB2 11.1 Activation Key Part Numbers
 - See IBM DB2 LUW 11.1 Knowledge Center



How do I download my DB2 license from Passport Advantage? The license files for DB2 Version 11.1 are shipped separately. This is done for convenience so that downloading the license file takes less time due to its small size. You will need to download the license activation key from Passport Advantage and then install it.

A detailed example of how to obtain the activation key for your DB2 product is shown at: <http://www-01.ibm.com/support/docview.wss?uid=swg21305417&myns=swgimgmt&mynp=OCSSEPPGG&mynp=OCSSEPPDU&mync=E>

The Activation key part numbers for DB2 11.1 products are located at the following website: <http://www-01.ibm.com/support/docview.wss?rs=0&uid=swg21305417>

3. Updating licenses

- To switch between DB2 11.1 Workgroup, Enterprise, and Advanced Edition licenses:
 - Cannot apply DB2 pureScale license over DB2 Advanced Enterprise Server Edition or DB2 Advanced Workgroup Server Edition terabyte license. If a DSF license exists then cannot apply DB2 Advanced Enterprise Server Edition or DB2 Advanced Workgroup Server Edition terabyte licenses. DB2 pureScale license cannot coexist with DB2 Advanced Enterprise Server Edition or DB2 Advanced Workgroup Server Edition terabyte licenses.
- Run **db2licm -l** to list all DB2 product licenses registered on system.
db2licm -l *product-identifier*
- Run **db2licm -r** to remove existing product license, feature license, and trial license.
db2licm -r *product-identifier*
- Run **db2licm -a** to apply new DB2 product license on system. If existing product license was not removed by running db2licm -r, it is automatically removed.
db2licm -a *product-identifier*
- Run **db2licm -l** to view newly applied DB2 product license information.
db2licm -l *product-identifier*
- Run **db2start** to restart instance, and for license compliance report to take effect for new license.
db2start

You can switch between editions in the DB2 11.1 - Workgroup, Enterprise, and Advanced Editions common image.

4. Post-installation tasks: DB2 Windows servers

- Add users to DB2ADMNS and DB2USERS user groups
- Update 32-bit DB2 instances to 64-bit instances
- **Validating DB2 copy: `db2val -a`**
- Set up notification and contact lists
- List DB2 copies and change default DB2 or default IBM database client interface copy after installation:
`db2swtch -l`
`db2swtch -d DB2_copy_name`
- Check for DB2 updates
- Install SSH Secure Shell Server component after installing DB2 product
 - Provides secure platform to run commands from remote machines
 - IBM SSH for Windows required for Optim Database Administrator
http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.qb.server.doc/doc/c0059310.html?lang=en



After installing DB2 database on Windows systems, you must complete a number of tasks.

- **Adding your user ID to the DB2ADMNS and DB2USERS user groups (Windows)**
 - After successfully completing a DB2 installation, you now have to add users to the DB2ADMNS or the DB2USERS groups for users that need to run local DB2 applications and tools on the machine. The DB2 installer creates two new groups. You can either use a new name or accept the default names. The default group names are DB2ADMNS and DB2USERS. This process is also applicable to instance based clients.
- **Updating your 32-bit DB2 instances to 64-bit instances (Windows)**
 - If you have more than one 32-bit DB2 copy on your system, you can convert them to 64-bit instances.
- **Validating your DB2 copy**
 - The `db2val` command ensures that your DB2 copy is functioning properly.
- **Setting up notification and contact lists**
 - For your DB2 database product to contact you or others regarding the status of your databases, set up notification and contact lists. If you did not do this during the DB2 database product installation, you can manually set up these lists.
- **Changing the default DB2 and default IBM database client interface copy after installation (Windows)**
 - The `db2swtch` command can be run from any DB2 copy or IBM data server driver copy. If no arguments, launches the utility in graphical mode. The `-l` displays a list of DB2 copies and IBM data server driver copies on the system. To switch the default DB2 copy to the name specified: `db2swtch -db2 -d DB2_copy_name-client`. The `-d` specifies the name of DB2 copy or IBM data server driver copy. To switch the default client interface copy to the name specified: `db2swtch -client -d name of DB2 copy or IBM data server driver copy-d DB2_copy_name`. To switch both default DB2 copy and client interface copy to the name specified: `db2swtch -d DB2_copy_name`.
- **IBM data server client connectivity using multiple copies**
 - When using multiple DB2 copies of DB2 database products or multiple data server driver copies, various options are available for applications to access the DB2 databases. Existing applications will continue to function properly.
- **Working with existing DB2 copies**
 - You can install and run multiple DB2 copies on the same computer. Each DB2 copy can either be at the same DB2 database product level or at a different DB2 database product level.

4. Post-installation tasks: DB2 Linux/UNIX servers

- List DB2 database products installed on system: **db2ls**
 - To list info about DB2 database products or features in particular installation path `q` parameter must be specified:
 - **db2ls -q -p -b baseInstallDirectory**
 - `q` specifies querying product or feature. Parameter is mandatory.
 - `p` specifies listing displays products rather than listing features.
 - `b` specifies installation directory of product or feature. Parameter is mandatory if not running command from installation directory.
 - Forgot which DB2 features you installed?
`/opt/IBM/db2/11.1/install/db2ls -q -a -b /opt/IBM/db2/11.1`
- Verify basic functions of DB2 copy by checking state of installation files, instance setup, and local database connections.
 - **Validate all instances for DB2 copy:** **db2val -a**
- Main menu entries for DB2 tools (Linux)
 - After installation, can add several DB2 tools to Main menu. See notes for details.



After installing DB2 database on Linux or UNIX systems, you must complete a number of tasks.

- **Listing DB2 database products installed on your system (Linux and UNIX)**

On supported Linux and UNIX operating systems, the `db2ls` command lists the DB2 database products and features installed on your system. With the ability to install multiple copies of DB2 database products on your system and the flexibility to install DB2 database products and features in the path of your choice, you need a tool to help you keep track of what is installed and where it is installed. On supported Linux and UNIX operating systems, the `db2ls` command lists the DB2 products and features installed on your system. The `db2ls` command can be found both in the installation media and in a DB2 install copy on the system. The `db2ls` command can be run from either location. The `db2ls` command can be run from the installation media for all products except IBM Data Server Driver Package. The `db2ls` command can be used to list:

- Where DB2 database products are installed on your system and list the DB2 database product level
- All or specific DB2 database products and features in a particular installation path.

The output that the `db2ls` command lists is different depending on the ID used:

- When the `db2ls` command is run with root authority, only root DB2 installations are queried.
- When the `db2ls` command is run with a non-root ID, root DB2 installations and the non-root installation owned by matching non-root ID are queried. DB2 installations owned by other non-root IDs are not queried.

Reference: Student Notes

The `db2ls` command is the only method to query a DB2 database product. You *cannot* query DB2 database products using Linux or UNIX operating system native utilities, such as `pkginfo`, `rpm`, `SMIT`, or `swlist`. Any existing scripts containing a native installation utility that you use to query and interface with DB2 installations must change. You *cannot* use the `db2ls` command on Windows operating systems.

- **Validating your DB2 copy**

- The `db2val` command ensures that your DB2 copy is functioning properly. The `db2val` tool verifies the core function of a DB2 copy by validating installation files, instances, database creation, connections to that database, and the state of partitioned database environments. This validation can be helpful if you have manually deployed a DB2 copy on Linux and UNIX operating systems using `tar.gz` files. The `db2val` command can quickly ensure that all the configuration has been correctly done and ensure that the DB2 copy is what you expect it to be. You can specify instances and databases or you can run `db2val` against all of the instances. The `db2val` command can be found in the `DB2-install-path\bin` and `sqlib/bin` directories.

Main menu entries for DB2 tools (Linux)

After installation, you can add several DB2 tools to the Main menu:

Check for DB2 Updates

Command Line Processor (CLP) and Command Line Processor Plus (CLPPlus)

First Steps

These DB2 tools can be added to main menu automatically or manually. Main menu entries are created automatically when any of the following DB2 commands are run. The `db2icrt` and `db2iupdt` commands must be run by root. The `db2nrcfg` and `db2nrupdt` are for non-root install and are run by the instance owner.

- `db2icrt`, `db2iupdt`, `db2nrcfg`, `db2nrupdt`.

To see the menu entries you might need to restart the desktop. The main menu entries are automatically removed when either of the following commands are run:

- `db2_deinstall` (only removes the menu entries for the DB2 non-root instance when the non-root installation is removed)
- `db2idrop`.

Student Notes: For Reference Only

You can manually create or remove main menu entries by running the following commands:

- `db2addicons` – to add menu entries
- `db2rmicons` – to remove menu entries.

Before running the `db2addicons` command, the DB2 instance environment must be set for the current user. The instance environment can be setup with:

Instance_HOME/sqllib/db2profile (for Bourne shell and Korn shell users), or
Instance_HOME/sqllib/db2chsrc (for C shell users), where *Instance_HOME* is the instance owner's home directory.

db2val - DB2 copy validation tool

- **-o** Specifies only installation files will be validated. If specified, -i, -a, -b, and -s are ignored. Validation of instance, database, and extended security not performed
- **-i instance_name** Specifies instance to validate. For multiple instances, specify parameter multiple times. On Windows, if not specified, current instance is default. On Linux/UNIX, parameter only be used by root users in root installation of DB2 copy.
- **-a** Validates all instances in DB2 copy. On Linux /UNIX, parameter only be used by root users in a root installation of a DB2 copy. Overrides parameter -i.
- **-b <db_name>** Validates database creation and connections to database specified. Only active DB2 instances will be validated. Parameter ignored for DB2 client and DB2 pureScale instances.
- **-t <trace_file>** Applies only to Linux/UNIX. Specifies full path and name of trace file.
- **-d** Deprecated. Use the -t parameter instead.
- **-s** Starts DB2 database manager for specified instance that is part of partitioned database environment.
- **-l <log_file>** Writes log to file name specified. Default on Linux/UNIX is /tmp/db2valxx.log and on Windows My Documents\DB2LOG\db2valxx.log.
- **-? | -h** Displays usage information

46

© 2016 IBM Corporation

Verifies basic functions of DB2 copy by checking state of installation files, instance setup, and local database connections.

You can use the db2val command only to validate installation files in a DB2 pureScale environment.

Instance validation requires one of the following authorities:

- On root copies, root authority is required on Linux and UNIX operating systems.
- SYSADM plus one of the following authorities:
 - Instance owner
 - Root access on Linux and UNIX operating systems, or Local Administrator authority on Windows operating systems

To validate the instance TEST1 and the database DATA1, run the following command:

```
db2val -i TEST1 -b DATA1
```

To validate all the instances for the DB2 copy, run the following command:

```
db2val -a
```

To validate only the DB2 installation files, run the following command:

```
db2val -o
```

5. OPTIONAL: First Steps interface - db2fs

- Manually start First Steps using **db2fs**:

- **\$ db2fs**

- Create sample database.
- Check for product updates.
- Learn about licensing and support.
- To uninstall single feature:

\$ /opt/IBM/db2/11.1/install/db2_deinstall -F FIRST_STEPS



- **Linux and UNIX**

- db2fs command located in *INSTHOME*/sqlib/bin directory, where *INSTHOME* is instance home directory. SYSADM authority required.
 - Alternatively, on Linux operating systems can start First Steps from Main Menu by selecting IBM DB2 > First Steps.

- **Windows**

- db2fs command located in *DB2DIR*\bin directory, where *DB2DIR* is set to location specified during DB2 installation.
 - Alternatively, can start First Steps from Start menu by selecting Programs > IBM DB2 > [*DB2 Copy Name*] > Set Up Tools > First Steps or from DB2 database product folder on desktop.

Launch the First Steps interface, which contains links to the functions that you need to begin learning about and using the DB2 product.

From First Steps you can:

- Create the sample database or your own database.
- Check for product updates.
- Learn more about DB2 database product licensing and support.

Product updates

Ensure that you are aware of the available product updates and enhancements for DB2 database products.

With the update service you can view a list of available DB2 database product updates, and learn details about DB2 database product updates.

db2ckupgrade - Check database for upgrade

db2ckupgrade <database-> -l filename

- In partitioned database, db2ckupgrade will check each database partition.
- On Linux/UNIX, install new DB2 copy and run from *DB2DIR*/bin directory where *DB2DIR* is location where DB2 copy is installed
- On Windows, insert DB2 product CD and run from db2\Windows\Utilities directory on CD
- SYSADM authority

db2ckupgrade -e -l filename

- e specifies all local cataloged databases are to be scanned

db2ckupgrade <database-> -l filename

- l filename specifies log file to keep list of errors and warnings



- Confirm running correct level of db2ckupgrade command. Ensure db2ckupgrade log file contains following text:

Version of DB2CKUPGRADE being run: Version 11.1

The **db2ckupgrade** command must be issued prior to upgrading the instance to verify that your databases are ready for upgrade. Backup all databases prior to upgrade, and prior to the installation of the current version of DB2 database product on Windows operating systems. Once the codebase has been installed you should then run the db2ckupgrade before upgrading the instance or databases. The db2ckupgrade fails to run against databases which are catalogued as remote databases.

The **db2ckupgrade** command verifies that a database can be migrated. In a partitioned database environment, the db2ckupgrade command will check each database partition.

Ensure that the log file for db2ckupgrade command contains the following text: Version of DB2CKUPGRADE being run: Version 11.1. This text confirms that you are running the correct level of the db2ckupgrade command.

SYSADM authority is required to execute. No database connection is required.

db2ckupgrade verifies following conditions are true:

- A database is not using raw logs.
- A catalogued database actually exists.
- A database is not in inconsistent state.
- A database is not in backup pending state.
- A database is not in restore pending state.
- A database is not in roll forward pending state.
- Tables are not in load pending state.
- Tables are not in redistribute pending state.
- That all table space container paths use the same mount point.
- That I/O write operations for database are not suspended
- No MQT's that depend on system views.
- Table spaces are in normal state.
- Database does not contain UDTs with name ARRAY, BINARY, CURSOR, DECFLOAT, ROW, VARBINARY, or XML.
- Database does not contain built-in DATALINK data type.
- Database does not have schema with name SYSPUBLIC.
- Database does not have orphan rows in system catalog tables that cause database upgrade to fail.
- Database enabled as HADR primary database allows successful connections.
- For V9.7, 10.1 and 10.5 Fix Pack 6 or earlier, HADR database role is not standby.
- Database is not enabled for XML Extender.
- If SYSCATSPACE is DMS table space and AUTORESIZE not enabled, SYSCATSPACE has at least 50% free pages of total pages.

FALSE
true

49

© 2016 IBM Corporation

A local database must pass all of these checks to succeed at the upgrade process.

The **db2iupgrade** command implicitly calls the **db2ckupgrade** command and specifies update.log as the log file for db2ckupgrade. The default log file created for db2iupgrade is /tmp/db2ckupgrade.log.processID. The db2iupgrade fails if the db2ckupgrade command finds that any of the above listed conditions are not true, and returns the DBI1205E error code. The user needs to resolve these errors before upgrading the instance.

db2chkupgrade verification (continued)

- db2ckupgrade writes warning message to log file (-l parameter) for any of following conditions:
 - Column names, routine parameter names, or variable names are called NULL.
 - Workload connection attributes contain asterisks (*).
 - Database enabled for DB2 WebSphere MQ functions.



For partitioned database environments, run the db2ckupgrade command before you issue the db2iupgrade command. The db2ckupgrade command will check all partitions and will return errors found in any partition. If you do not check whether all database partitions are ready for upgrade, subsequent database upgrades could fail even though the instance upgrade was successful.

db2ckupgrade – SQL0430N error

SQL0430N – Someone has removed the fenced user id or fenced userid does not have adequate permissions on instance

[IBM[[CLI Driver][DB2/AIX64] SQL0430N User defined function "SYSPROC.REG_LIST_VARIABLES" (specific name "REG_LIST_VARIABLES") has abnormally terminated SQLSTATE=38503



1. Check if DB2 fenced user id exists:

\$ ls -al \$HOME/sqllib/.fenced

(where \$HOME is instance home directory)

If owner of .fenced file is numeric value, then user was dropped.

2. Root resolve problem by issuing db2iupdt from \$DB2DIR/instance with -u option and valid fenced userid (where \$DB2DIR is installation directory for DB2):

db2iupdt -u db2fenc1

When migrating a DB2 database, the db2ckupgrade command could produce a SQL0403N error which means that someone removed the fenced user id or the fenced user id does not have adequate permissions on the instance.

Check if DB2 fenced user id exists:

ls -al \$HOME/sqllib/.fenced (where \$HOME is instance home directory)

If owner of .fenced file is number then represents user that was dropped.

Resolve problem by issuing db2iupdt as root from \$DB2DIR/instance

(where \$DB2DIR is installation directory for DB2) with -u option and valid fenced userid.:

#db2iupdt -u db2fenc1

db2ckupgrade or db2iupgrade -b option

- > **PROBLEM:** db2ckupgrade or db2iupgrade take long time to complete.
 - > Additional check for tables in load pending state added to db2ckupgrade tool.
 - > If large number of tables, indexes, data, check can take long time.

- > **SOLUTION:** Only use if 0 rows returned!

```
SELECT tabname FROM SYSIBMADM.ADMINTABINFO
WHERE LOAD_STATUS IS NOT NULL
```

0 Records returned

- > **Internal -b option added since DB2 10.5 Fixpack 3 and DB2 10.1 Fixpack 3 bypasses check for tables in load pending state:**

```
db2ckupgrade <dbname> -b -l <mylog.log>
db2iupgrade -b <myinstance>
```



Remember: db2iupgrade calls db2ckupgrade!

The db2ckupgrade or db2iupgrade tools may take a long time to complete when upgrading. An additional check for tables in load pending state can take a long time if a database has a large number of tables, defined indexes and hosts a considerable amount of data. A query against SYSIBMADM.ADMINTABINFO, may be used to obtain this information from the database.

To diagnose the problem, looking at the db2diag.log file. It may show that the db2ckupgrade tool is spending a lot of time using the sqlmCkTableLoadPending() function.

The following example from db2diag.log file shows gap of 2 hours between beginning and end message for the sqlmCkTableLoadPending function:

2016-04-16-09.44.55.802740+000 I161183A335 LEVEL: Warning

PID : 5178038 TID : 1 PROC : db2ckupgrade64_exe

INSTANCE: db2inst NODE : 000

APPID : *N0.db2inst.160416094415

EDUID : 1

FUNCTION: <0>, <0>, <0>, probe:4138

DATA #1 : <preformatted>

DB2CKUPGRADE: Begin: sqlmCkTableLoadPending

Student Notes: For Reference Only

2016-04-18-09.46.20.491662+000 I64434131A309 LEVEL: Warning

PID : 5178038 TID : 1 PROC :

db2ckupgrade64_exe

INSTANCE: db2inst NODE : 000

APPID : *N0.db2inst.160416094415

EDUID : 1

DATA #1 : <preformatted>

DB2CKUPGRADE: End: sqlmCkTableLoadPending with rc = 0

While the issue actually occurs during the execution of the db2ckupgrade or db2iupgrade tools, only one initiate see the begin message for the sqlmCkTableLoadPending function.

Additional active SQL monitoring will show following statement running:

```
"select tabname from SYSIBMADM.ADMINTABINFO where load_status is not NULL"
```

You can use a temporary work around for the issue. Starting with the 10.1 Fix Pack 3 and 10.5 Fix Pack 3 versions of the DB2 product, the db2ckupgrade and db2iupgrade tools have an internal option , -b, which bypasses the check for tables in load pending state

Disk space requirements (1)

• System catalog table space (SYSCATSPACE)

- Increase to twice the total of used space.
- Free pages should be \geq used pages.
- System catalog table space is required for both old and new database catalogs during upgrade.



• Temporary table space (TEMPSPACE1)

- Increasing total size to twice total size of system catalog table space.

```
SELECT
SUBSTR(TBSP_NAME,1,15) NAME, TBSP_TYPE TYPE,
TBSP_AUTO_RESIZE_ENABLED AUTO_RESIZE, TBSP_NUM_CONTAINERS
CONTAINERS, TBSP_TOTAL_PAGES TOTAL_PGS, TBSP_USED_PAGES USED_PGS,
TBSP_FREE_PAGES FREE_PGS, TBSP_MAX_SIZE MAX_SZ, TBSP_PAGE_SIZE PG_SZ
FROM SYSIBMADM.TBSP_UTILIZATION
WHERE TBSP_CONTENT_TYPE IN ('ANY','SYSTEMP')
```

You must be aware that the upgrade process requires additional disk space. Ensure that you have enough free disk space to complete this process successfully. The above disk space recommendations are applicable for upgrading to DB2 11.1.

Ensure that you have sufficient free space on the system catalog and the system temporary table spaces for the databases that you are upgrading. System catalog table space is required for both old and new database catalogs during upgrade. The amount of free space required varies, depending on the complexity of the database, as well as on the number and size of database objects.

To increase the amount of free space on your automatic storage table spaces, you can increase the space on the current storage paths or add a new storage path.

To increase the amount of free space on your System Managed Space (SMS) table spaces, free sufficient disk space on the corresponding file systems or increase the size of your file systems if you are using a volume manager.

To increase the amount of free space on your Database Managed Space (DMS) table spaces, you can increase the size of existing containers. You can also add additional containers although this might trigger data rebalancing. You can reduce the size of the containers after upgrade.

Disk space requirements (2)

- **Instance Upgrade Trace File written to /tmp for LINUX and UNIX**
 - Ensure 5GB of free space in /tmp directory.
- **Database log file space**
 - Upgrade changes to system catalog objects are single UOW.
 - Set LOGSECOND to twice current value of LOGPRIMARY and LOGSECOND if file system with log files has enough disk free space.
 - If already have large log file space, might not be necessary.
 - **Only need to increase log space on catalog partition.**
 - **Must update these values before upgrade the instance to DB2 10.5**
 - Optional: Consider infinite logging (LOGSECOND=-1)

db2 GET DB CFG FOR sample | grep '(LOG[FPS])' | tee logsize.txt

db2 UPDATE DB CFG FOR sample using LOGSECOND
(current_value of LOGPRIMARY + current_value of LOGSECOND)



Instance upgrade trace file is written to /tmp for Linux and UNIX. Ensure at least 5GB of free space in /tmp directory.

You must update these LOGSECOND values before you upgrade the instance to DB2 11.1, because you will not be able to update these DB CFG parameters until issue the UPGRADE DATABASE command. If this command fails because there is insufficient log file space, then you can set these database configuration parameters to higher values and then re-issue the UPGRADE DATABASE command.

The new database configuration parameter settings for log space can be restored to their original value after the upgrade is complete.

Pre-upgrade tasks for DB2 servers

- ¹ If use distributed transactions, ensure databases do not contain any indoubt transactions by using LIST INDOUBT TRANSACTIONS to get list of indoubt transactions and interactively resolve any indoubt transactions.
- ² Verify databases are ready for DB2 upgrade to identify any problems before actual upgrade. Must resolve them before proceed with upgrade.
- ³ Optional: **Stop HADR on primary and standby databases** or may **upgrade without re-initializing standby database if on DB2 10.5 Fix Pack 7 or later.**
- ⁴ **Back up databases to be able to upgrade them to a new upgraded system or restore them in the original pre-upgrade system.**
- ⁵ **Back up configuration and diagnostic information to have record of current configuration that can compare with configuration after upgrade. Use this information to create new instances or databases using same configuration that had before upgrade.**
- ⁶ Archive all of DB2 log files, either for SQL or Q replication if log files needed by Capture or Q Capture, or for HADR if needed to create standby database.
- ⁷ **Review disk space requirements to ensure have enough free disk space, system temporary table space and log space for the upgrade and increase table space and log file sizes if necessary.**
- ⁸ Windows only: If obtained customized code page conversion tables from the DB2 support service, need to backup all files in DB2OLD\conv directory where DB2OLD is the location of your existing pre-DB2 11.1 copy.
- ⁹ Linux only: Change raw devices to block devices.
- ¹⁰ Optional: Upgrade DB2 server in test environment to identify upgrade issues and verify applications, scripts, tools and routines work as expected before upgrading DB2 server in production environment.
- ¹¹ **If diaglevel is 2 or less, set diagnostic error capture level to 3 or higher.**
- ¹² **Take DB2 server offline for upgrade.**
- ¹³ Refresh data in existing MQTs. All MQTs that depend on system views are dropped during database upgrade. After upgrade, must refresh data in existing MQTs by using REFRESH TABLE.
- ¹⁴ Raw logs are not supported in DB2 11.

56

© 2016 IBM Corporation

If you use distributed transactions involving DB2 databases, ensure that the databases to be upgraded do not contain any indoubt transactions by using the LIST INDOUBT TRANSACTIONS command to get a list of indoubt transactions and to interactively resolve any indoubt transactions. In DB2 11.1, all significant upgrade events are logged in the **db2diag** log files when the **diaglevel** database manager configuration parameter is set to 3 (default value) or higher. If this parameter is set to 2 or less, set this parameter to 3 or higher before upgrade.

Backing up DB2 server configuration and diagnostic info

Collect information from DB2 servers by running `db2support` command for all databases going to upgrade in all your instances. Collect information about database system catalog, DB CFG and DBM CFG settings, DB2 registry variables settings, explain table data, and diagnostic information required by DB2 support in case of problems.

```
db2support output-directory -d database-name -cl 0
```

- Keep zip for several months after complete upgrade or conversion to DB2 pureScale to help resolve performance issues with new release.

Back up information about all packages for applications associated with each database.

```
db2 LIST PACKAGES FOR ALL SHOW DETAIL > /upgrade/sample_pckg.txt
```

If enabled audit facility, back up audit cfg for each instances:

```
db2audit describe > audit_instance-name.cfg
```

Back up all external routines.

```
cp -R $INSTHOME/sqllib/function $INSTHOME/routine_backup
```

Optional Steps: `db2support` HTML report includes following info.

** Use `GET DBM CFG` to back up settings for each instance:

```
db2 GET DBM CFG > dbm_instname.cfg
```

** Use either `db2look` or `GET DB CFG` to back up settings for each database.

If not same settings, back up DB CFG for each database partition.

```
db2 CONNECT TO database_alias
```

```
db2 GET DB CFG FOR database_alias SHOW DETAIL > db_database_alias.cfg
```

```
db2look -sample -printdbcfg -o sample_cfg.db2 (10.5 Fix Pack 4 or >)
```

** Use `db2look` to save additional information.

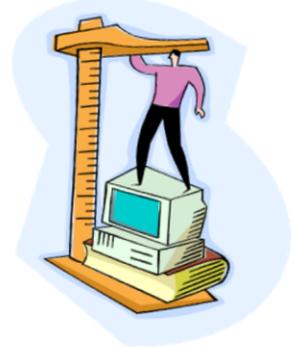
```
db2look -d sample -e -o sample_tbs.db2 -l -x
```

** Use `db2set` command to back up DB2 profile registry variables settings:

```
db2set -all > reg_instname.txt
```

If set DB2 environment variables, use appropriate system command to list environment variables values.

```
set | grep DB2 > env_instname.txt (UNIX example)
```



** Optional: Contained in
db2support zip file

Backing up your settings for database and database manager configuration parameters before DB2 server upgrade allows you to verify DB2 server behavior after upgrade and to re-create instances and databases. In addition, you can collect information from your DB2 servers about the database system catalogs, DB2 registry variables settings, explain table data, and diagnostic information that can help in problem determination if you encounter any post-upgrade differences in the database manager behavior or performance.

The `db2support -cl 0` parameter collects the database system catalog, database and database manager configuration parameters settings, DB2 registry variables settings. The information collected is stored in the `db2support.zip` compressed zip file under the output directory. A summary report in HTML format is included. In the `db2supp_opt.zip` file that is also included, you should check the `optimizer.log` file to verify that the collection of information was performed successfully. Keep this zip file for several months after you complete the upgrade, or conversion to DB2 pureScale. The information in the zip file can help in quickly resolving any performance issues with the new release.

The `LIST PACKAGES ... FOR SCHEMA` clause allows you to list all packages for a specific schema, if your application has several schemas you need to repeat this command for each schema name or use the `FOR ALL` clause.

When backing up external routines, `INSTHOME` is set to home directory of instance owner. If you have specified a full path that is not under the default routines path when you created your external routines in the database, you must ensure the existing libraries remain on their original location.

The `SHOW DETAIL` clause on `GET DB CFG` displays the values calculated by the database manager when configuration parameters are set to `AUTOMATIC`.

When possible, use the output from the `set` command and run the `db2set` command to set these environment variables as registry variables in the DB2 profile registry.

Back up databases before AND after upgrade

- Backup used for reversing instance/database upgrade.
- **REMEMBER: After upgrade instances to 11.1, cannot backup databases until upgrade them!**
- **Can optionally take ONLINE BACKUP if all of following are true:**
 - Updating database "in place".
 - Single partition DB2 10.5 Fix Pack 7 or pureScale 10.5 Fix Pack 7 or later.
 - Ensure have full database backup image and log files from pre-DB2 11.1 release and log files from 11.1 after upgrade completes.
 - Ability to roll-forward through database version upgrades
- **OFFLINE BACKUP required if any of the following are true:**
 - Restoring pre-DB2 11.1 backup into new DB2 11.1 instance
 - Multiple partitioned database. **Must back up all database partitions.**
 - Not single partition DB2 or pureScale 10.5 Fix Pack 7 or later
 - If performed full online or offline database backup recently and cannot perform another one before upgrading, can perform incremental offline database backup instead if TRACKMOD=ON.
 - If activated and configured DB2 Advanced Copy Services (ACS), can use USE SNAPSHOT parameter to perform snapshot backup. Can only restore snapshot backup to instance of same version. Cannot use snapshot backup to upgrade to new version.



Before and after the upgrade process to DB2 11.1, it is strongly recommended that you perform a full database backup. If an error occurs during the upgrade process, you need full database backups to recover and upgrade your databases.

After you upgrade your instances to DB2 11.1, you cannot backup databases until you upgrade them.

Instance upgrade - db2iupgrade

- Call instance upgrade explicitly with **db2iupgrade** command
- Call implicitly when install DB2 11.1 on Windows and select **Work with Existing option** and choose a pre-11.1 copy with upgrade action
- db2iupgrade does following things:
 - Upgrades existing instance to new instance under DB2 11.1 copy.
 - Upgrades instance profile registry variables. **Global profile registry variables are not upgraded.** 
 - Upgrades dbm cfg file.
 - Sets jdk_path dbm cfg parameter.
 - Upgrades db2audit.cfg audit cfg file when audit facility enabled.
 - Uses SSLconfig.ini SSL configuration file to set new dbm cfg parameters to corresponding SSL parameter value in this file and upgrades instance profile registry setting DB2COMM=SSL.
 - For successful instance upgrade, all files must exist for all instances and all files must have write access granted.

59

© 2016 IBM Corporation

Only DB2 Enterprise Server Edition instances (instance type ese) can be upgraded using the db2iupgrade command. The db2iupgrade command calls the db2ckupgrade command and specifies update.log as the log file for db2ckupgrade. The default log file created for db2iupgrade is /tmp/db2ckupgrade.log.processID. Verify that local databases are ready for upgrade before upgrading the instance. The log file is created in the instance home directory for Linux and UNIX operating systems or in the current directory for Windows operating systems. The instance upgrade will not continue if the db2ckupgrade command returns any errors.

If you use the db2iupgrade command to upgrade a DB2 instance from a previous version to the current version of a Linux/UNIX DB2 database system, the DB2 Global Profile Variables defined in an old DB2 database installation path will not be upgraded over to the new installation location. The DB2 Instance Profile Variables specific to the instance to be upgraded will be carried over after the instance is upgraded. If you are using the su command instead of the login command to become the root user, you must issue the su command with the - option to indicate that the process environment is to be set as if you had logged in to the system using the login command. You must not source the DB2 instance environment for the root user. Running the db2iupgrade command when you sourced the DB2 instance environment is not supported.

On Windows operating systems, this command is located in the DB2PATH\bin directory, where DB2PATH is the location where the DB2 copy is installed. To move your instance profile from its current location to another location, use the /p option and specify the instance profile path. Otherwise, the instance profile will stay in its original location after the upgrade.

Upgrading to DB2 11.1 Instance – db2iupgrade

- . **BACKUP DATABASE** before upgrading!
- . db2stop force
- . db2 terminate
- . **cd /opt/IBM/db2/11.1** (code install path)
- . **db2iupgrade –u fencedid inst1**
 - Root access on Linux/UNIX or Local Administrator on Windows
 - Calls **db2ckupgrade** with **–e** and **–l db2ckupgrade.log**
 - Fails if db2ckupgrade fails



```
>>-db2iupgrade--t-----t--t-----t--t-----t--t-----t--InstName-><
      '-d-' '-k-' '-a--AuthType-' '-u--FencedID-'
```

-d turns debug mode on. Use when instructed by DB2 Support.

-k keeps pre-upgrade instance type. If not specified instance is upgraded if supported.

-a specifies the AUTHENTICATION type. Default is SERVER.

-u specifies user id which fenced UDF and stored procedures run.

-j configures DB2 Text Search server's service name and port number. See student notes for details.

For Windows: **cd DB2PATH\bin directory** (code install path)
db2iupgrade /d inst1

60

© 2016 IBM Corporation

Before starting the instance migration, the instance must be stopped. You must be logged in as root to perform the majority of tasks. It is important that the DB2 runtime instance environment, normally established using the db2profile script, has not been set. If it is, the instance migration commands will not work. The db2iupgrade command upgrades an instance to a DB2 copy of the current release from a DB2 copy of a previous release. The DB2 copy from where you are running the db2iupgrade command must support instance upgrade from the DB2 copy that you want to upgrade.

On Linux and UNIX, this command is located in the DB2DIR/instance directory, where DB2DIR represents the installation location where the new release of the DB2 database system is installed. This command does not support instance upgrade for a non-root installation. On Windows, this command is located in the DB2PATH\bin directory, where DB2PATH is the location where the DB2 copy is installed. To move your instance profile from its current location to another location, use the /p option and specify the instance profile path. Otherwise, the instance profile will stay in its original location after the upgrade.

The db2iupgrade command calls the db2ckupgrade command with the -not1 parameter and specifying db2ckupgrade.log as the log file. Verify that local databases are ready for upgrade before upgrading the instance. The -not1 parameter disables the check for type-1 indexes.

A log file giving details of what took place is written to the instance home directory (UNIX-like systems) or to the current directory path (Windows). If any errors are reported, check this log file for further details.

-j "TEXT_SEARCH"

Configures DB2 Text Search server using generated default values for service name and TCP/IP port number. Cannot be used if the instance type is client.

-j "TEXT_SEARCH, servicename"

Configures DB2 Text Search server using provided service name and automatically generated port number. If service name has port number assigned in services file, it uses assigned port number.

-j "TEXT_SEARCH, servicename, portnumber"

Configures DB2 Text Search server using provided service name and port number.

-j "TEXT_SEARCH, portnumber"

Configures DB2 Text Search server using default service name and provided port number. Valid port numbers must be within 1024 - 65535 range.

Validating Instance Upgrade

Review **db2iupgrade log file** written to Windows current path or UNIX/Linux instance home directory. Review **db2diag** and **administrative notification logs**.

- Switch to the instance owner and issue:

 **db2start**
db2level

DB21085I This instance or install (instance name, where applicable: "inst461") uses "64" bits and DB2 code release "SQL11010" with level identifier "0601010E". Informational tokens are "DB2 11.1.0.0", "s130528", "LINUXAMD64105", and Fix Pack "0". Product is installed at "/opt/ibm/db2/11.1".

- On Windows, db2level shows DB2 copy name.
- To display the same information for DB2 pureScale member, use:
db2level -localMember



The db2level output shown above was ran on a Linux 64-bit operating system with DB2 11.1 and no fix packs installed.

The db2iupgrade log file is created in the instance home directory for Linux and UNIX operating systems or in the current directory for Windows operating systems. The instance upgrade will not continue if the db2ckupgrade command returns any errors.

After upgrade, check output of db2diag and administrative notification logs.

If the db2iupgrade command completes successfully, switch to the instance owner and issue the **db2start** to restart the instance. To confirm you are now running DB2 11.1 issue the **db2level** command.

The **-localMember** parameter specifies the current version and service level for the member where the db2level command is being issued.

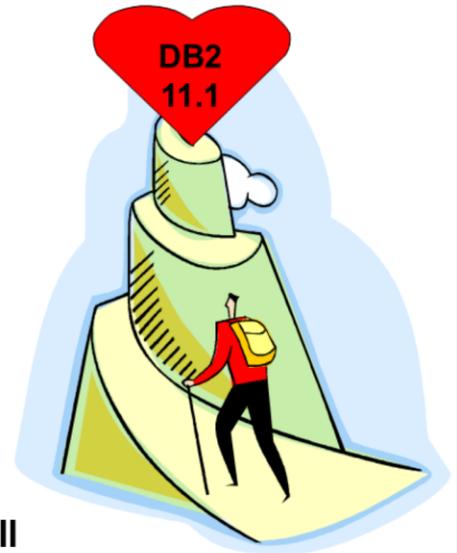
Database directory upgrade

- During first time access, database directory is implicitly upgraded if necessary.
- Database directory is accessed when issue:
 - **LIST DATABASE DIRECTORY**
 - **UPGRADE DATABASE**
- Obtain list of all local databases that must be upgraded:

```
db2 list db directory > db-dir-before.txt
```

DBY
DBX

**Must upgrade all
databases
in upgraded instance**



The database directory is implicitly upgraded if necessary during first time access.

After the instance upgrade has been completed, all databases need to be upgraded to the latest version.

Obtain a list of all local databases using the **list database directory** command.

Database upgrade no longer renames log files

- When upgrading from DB2 9.7, 10.1 or 10.5, database log files will no longer be renamed from .LOG to .MIG.
- For 10.1 and 10.5, any unarchived files prior to upgrade will be archived from uplevel and will not require any action.
- For 9.7, any unarchived files will not be archived, and will be left in the old log directories. Users will have to back up, archive, and manage the files manually.
- db2cklog tool can assist in manual management of log files to confirm what logs are from previous DB2 versions. If recovery log file not from current version, then a DBT warning message returned.



db2cklog CHECK 80 to 94 ARCHLOGPATH <path where logfiles stored>

- Default is current directory

Reversing Upgrade: Since no longer rename log files from previous DB2 versions, if need to reverse upgrade, before dropping database, log files in active log directories should be manually saved before proceeding.

Prior to 11.1, for recoverable databases, UPGRADE DATABASE renames all log files in active log path with the extension .MIG. After upgrade databases successfully, you deleted all S*.MIG files. In DB2 11.1, a database upgrade no longer renames log files.

Smart Idea - Before upgrade, clean up diagpath

- Rename or delete db2diag log files so new files are created.
- Remove or move to another directory any existing dump files, trap files, and alert log files in diagnostic path indicated by diagpath.
- Now files only contain information about database upgrade process.
- Helps to isolate and understand any problem that might occur during database upgrade.



Rename or delete the db2diag log files so that new files are created. Also, remove or move to another directory any existing dump files, trap files, and alert log files in the directory indicated by the diagpath parameter. By doing this, the files only contain information about the upgrade process that helps you to isolate and understand any problem that might occur during database upgrade.

Upgrading Local Databases to DB2 11.1

db2 UPGRADE DATABASE *database_alias*

- Instance owner authority required
- **REBINDALL** rebinds all packages during upgrade
 - Alternately, rebind all valid and invalid packages after database upgrade:
 - **db2rbind *database-name* -l *logfile* all**

db2 UPGRADE DATABASE DBY -REBINDALL db2 UPGRADE DATABASE DBX -REBINDALL



If error occurs during upgrade, may need to issue **TERMINATE**

```
db2 CONNECT to <alias>
db2 GET DB CFG FOR alias SHOW DETAIL > alias.cfg
db2look -alias -printdbcfg -o alias.cfg (10.5 Fix Pack 4 or >)
```

```
db2 LIST TABLESPACES SHOW DETAIL
```

```
> <dbname>-tsp-after.txt
```

```
db2 LIST PACKAGES FOR ALL SHOW DETAIL
```

```
> <dbname>-pkg-after.txt
```

```
db2 CONNECT RESET
```

```
...
```

Compare before and after txt files.

65

© 2016 IBM Corporation

The **db2ckupgrade** command must be issued prior to upgrading the instance to verify that your databases are ready for upgrade. The **db2iupgrade** command implicitly calls the **db2ckupgrade**. Backup all databases prior to upgrade, and prior to the installation of the current version of DB2 database product on Windows operating systems.

The **upgrade database** command will only upgrade a database to a newer version, and cannot be used to convert an upgraded database to its previous version. The database must be cataloged before upgrade.

If an error occurs during upgrade, it might be necessary to issue the **TERMINATE** command before attempting the suggested user response. For example, if a log full error occurs during upgrade (SQL1704: Database upgrade failed. Reason code "3"), it will be necessary to issue the **TERMINATE** command before increasing the values of the database configuration parameters **LOGPRIMARY** and **LOGFILSIZ**. The CLP must refresh its database directory cache if the upgrade failure occurs after the database has already been relocated (which is likely to be the case when a "log full" error returns).

After each database has been upgraded repeat the statistics gathering as above and compare the two sets of information. With the exception of the system packages, the two sets of information should be identical.

The **REBINDALL** option specifies that a **REBIND** of all packages is performed during upgrade. Performing the **REBINDs** automatically during the upgrade ensures that the step is not missed and will help ensure that other applications are not started before the **REBINDs** are complete.

Upgrading partitioned database environments

- In configurations using multiple database partitions, **roll forward through a database upgrade is not supported.**
- Install DB2 11.1 on all database partition servers, upgrade the instances and then upgrade the databases.
- Use only “Install New” option in “Install a Product” panel to install DB2 11.1.
 - If choose to upgrade action when select Work with Existing option on Install a Product panel, installation fails.
- Perform a full **offline** backup for all database partitions. Verify databases are ready for upgrade, and perform any other pre-upgrade tasks that apply.
- Upgrade each instance on database partition server that owns the instance. Run UPGRADE DATABASE on catalog partition.
- Catalog partition must be available when issue UPGRADE DATABASE regardless on what database partition issue command is issued.
- If any database partitions are not available, these database partitions are not upgraded.

66

© 2016 IBM Corporation

Upgrading partitioned database environments requires that you install DB2 11.1 as a new copy in all database partition servers, upgrade the instances and then upgrade the databases.

Ensure that you have root access on Linux and UNIX operating systems or Local Administrator authority on Windows.

Ensure that you have SYSADM authority.

How to check partitioned database is consistent during upgrade operation?

- **If see following during upgrade:**

DBT5517N The db2ckupgrade utility failed because the database is in an inconsistent state.
DBT5529W The db2ckupgrade utility did not complete successfully. The database cannot be upgraded.

- **If 'All committed transactions have been written to disk = YES' , the partitions are consistent.**

```
db2_all`db2 get db cfg | grep -i "All committed transactions have been written to disk"
```

- **Check Database Consistency using db2pd:**

```
db2pd -dbpartitionnum 0 -dbcfg -db dbname | grep "Database is consistent"
```

- **In case database partition is not consistent , issue 'restart database' on that partition and check if state comes back to consistent using above commands.**

The above information shows how to check to see if multiple partition database is consistent during an upgrade operation.

HADR Upgrade Enhancements

- Eliminates need to stop HADR for upgrade and re-initialization of HADR standby after upgrade single-partitioned DB2 with single standby or multiple standbys.
 - Reduces window where no standby database exists and eliminates cost of sending backup image to standby site for re-initialization.
 - Re-initialization of standby is still option, but not recommended option.
- Requires primary and standby databases to both be at **DB2 10.5 Fix Pack 7 level or greater**.
- Requires that one system/instance be identified to contain all primary databases and second system/instance be identified to contain all standby databases.
 - Where necessary, issue graceful takeover to achieve this layout.
- Maintains database role and relies on normal log shipping and log replaying.
- Requires that primary shipped all log data to standbys and standbys have replayed all log data received.
 - **DB2 11.1 standby cannot replay log data from earlier DB2 versions.**
 - Enforced through db2ckupgrade run during db2iupgrade that does database consistency checking. **DO NOT BYPASS db2ckupgrade!**

For single-partition DB2 Enterprise Server Edition users upgrading from DB2 10.5 Fix Pack 7, or later, high availability disaster recovery (HADR) environments can now be upgraded without the need to reinitialize the standby database after performing an upgrade on the primary database. Reinitialization of the standby is still an option if the user wants, but is no longer the recommended option.

The supported HADR upgrade procedure requires the primary and standby databases both to be at a minimum of DB2 10.5 Fix Pack 7 level. Any other prior level is not supported and attempts to use the documented procedure fails. For these prior level databases, the standby must be reinitialized by a backup image that is shipped from the primary after upgrade. The HADR upgrade procedure requires that one system/instance be identified to contain all primary databases and the second system/instance be identified to contain all standby databases. Where necessary, issue graceful takeover to achieve this layout of primary and standby databases.

DB2 pureScale environments now support SYNC and NEARSYNC for DB2 high availability disaster recovery (HADR). This enhancement combines the continuous availability of the DB2 pureScale Feature with the robust disaster recovery capabilities of HADR.

Upgrade without re-initializing standby database - Two new DB2 Log Records introduced

To support ability to roll forward (replay) through database upgrade, two new propagatable log records have been introduced.

1. Database Migration Begin Log Record

- Written to mark start of database upgrade.

2. Database Migration End Log Record

- Written to mark successful completion of database upgrade.

▪ Single-partition databases upgrading from DB2 10.5 Fix Pack 7 or later HADR environments can now upgrade without need to re-initialize standby database after performing upgrade on primary database.

- DB2 pureScale upgrade from DB2 10.5 (all Fix Packs) still required to stop HADR before upgrading and re-initialize standby post-upgrade.



For single-partition databases upgrading from DB2 10.5 Fix Pack 7, or later, high availability disaster recovery (HADR) environments can now be upgraded without the need to re-initialize the standby database after performing an upgrade on the primary database.

DB2 pureScale users upgrading from DB2 10.5 (all Fix Packs) still required to stop HADR before upgrading and reinitialize standby post-upgrade.

Upgrading DB2 servers without automated HADR and WITH Re-initialization of Standby

- Consider **logbufsz**, **logfilsiz**, **logprimary**, **logsecond**, **locklist** & **maxlock** changes
- Backup upgraded Primary Database
- Restore on Standby
- Reinitialize HADR

Primary db2 stop hadr on db dbx

Standby db2 deactivate database dbx

Standby db2 stop hadr on db dbx

Primary UPGRADE INSTANCE & DATABASE

Primary db2 backup db dbx online to ...

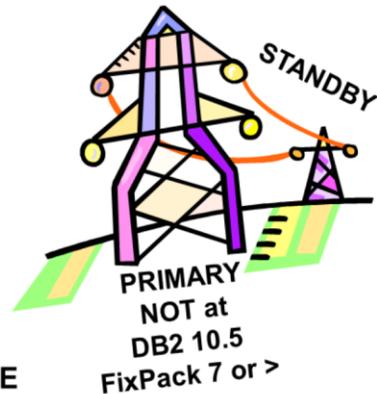
Standby db2iupgrade -u <fencedid> <standbyinstance>

move log files from previous DB2 version out of log path of new DB2 version

Standby db2 restore db dbx from ...

Standby db2 start hadr on db dbx as standby

Primary db2 start hadr on db dbx as primary



Upgrading a primary database to DB2 11.1 changes the database role from primary to standard.

Upgrading standby databases to DB2 11.1 is not supported because these databases are in roll forward pending state. Because of these restrictions, upgrading an HADR environment to DB2 11.1 requires that you stop HADR, upgrade your DB2 server where the primary database resides, and then re-initialize HADR. The primary database is taken out of HADR (its status is changed from “primary” to “standard”).

Before you activate your standby database, you must move the log files from the previous version of DB2 out of the log path of the new version of DB2. If you do not move the log files, DB2 might try to use the old files and fail to initialize.

Rolling Upgrade Procedure for TSAMP Automated HADR environment resources:

- White Paper:
<http://www-01.ibm.com/support/docview.wss?uid=swg21586363>

Upgrading DB2 HADR Servers with One Standby (1 of 3) Maintaining HADR role and without reinitializing HADR pair

Before You Begin:

- Review the system requirements for HADR: **db2pd -db <dbalias> -hadr**
 - *Both primary and standby database should be on same DB2 10.5 Fix Pack (7 or later).*
- If have two HADR databases in same instance, perform role switch (if necessary) so both primaries are on same system during upgrade: **db2 TAKEOVER HADR ON DB <dbalias>**
- Ensure you are familiar with steps involved in upgrading DB2 instance and DB2 databases.
- Ensure both primary's log shipping functionality and standby's log replaying functionality is working properly before running db2ckupgrade: **db2pd -db <dbalias> -hadr**
- If using Reads on Standby feature, ensure DB CFG **logindexbuild=on** so index recreation done during upgrade is sent to standby for replay.
 - Allows read connections to resume post upgrade on standby.
- In case of failures during HADR upgrade procedure, ensure familiar with "Dealing with failures while upgrading DB2 servers in HADR environments (10.5 Fix Pack 7 or later)".
- All DB2 upgrades, hardware upgrades, and software upgrades should be implemented in test environment before being applied to production system.

This procedure can be applied to HADR databases in Enterprise Server Edition environments that use single standby or multiple standby configurations. Use this procedure in a high availability disaster recovery (HADR) environment when you upgrade your single-partition DB2 Enterprise Server Edition 10.5 Fix Pack 7 or later databases to DB2 11.1. Both the primary and standby database should be using the same fix pack level of This procedure maintains the database role and relies on normal log shipping and log replaying characteristics common to HADR functionality. The procedure avoids the need to stop HADR for upgrade and avoids the need to reinitialize HADR. This reduces the window where no standby database exists and eliminates the cost of sending a backup image to the standby site for re-initialization.

Upgrading DB2 HADR Servers with One Standby (2 of 3)

- **PRIMARY:** Monitor HADR databases to reduce chance of upgrade failures.
 - Ensure primary log shipping and standby log replay are not lagging and functioning.
 - `db2pd -db <dbalias> -hadr` or `SELECT ... MON_GET_HADR ...`
- **STANDBY:** Verify `HADR_REPLAY_DELAY=0`. Ensures standby log replay position can catch up to primary in reasonable amount of time.
- **PRIMARY:** Ensure log shipping is completed on primary databases and data is transferred to standbys by deactivating primary databases:


```
db2 DEACTIVATE DATABASE <dbalias>
```
- **PRIMARY:** Ensure applications do not connect so no new log data generated and standbys log replay position eventually matches primary's log shipping position: `db2stop`
- **PRIMARY:** Upgrade DB2 instance: `db2iupgrade <instancename>`
 - `db2iupgrade` calls `db2ckupgrade` to verify primary database is ready for upgrade. **DO NOT BYPASS db2ckupgrade with -b option!**
 - `db2iupgrade` does not run if `db2ckupgrade` reports errors. Check log file if encounter errors.
 - For HADR primary database from 10.5 Fix Pack 7 or later, `db2ckupgrade` verifies that valid standby database can be connected to primary database.
 - Once connection established, log shipping begins and ships any pending log data, if necessary.
 - `db2ckupgrade` verifies that log shipping position on primary matches log replay position on standby
- **STANDBY:** Ensure log shipping is completed on primary databases and data is transferred to standbys:


```
db2 DEACTIVATE DATABASE <dbalias>
```
- **STANDBY:** `db2stop`



The procedure depends on the DB2 10.5 Fix Pack 7 or later primary database having shipped all log data to the DB2 10.5 Fix Pack 7 or later standby databases and the standby databases having replayed all log data received. The DB2 11.1 standby database cannot replay log data from earlier DB2 versions. The procedure enforces this restriction through `db2ckupgrade` that is run during `db2iupgrade` or any `db2ckupgrade` invocation that establishes an exclusive connection to the database to do proper database consistency checking.

This procedure can be applied to HADR databases in Enterprise Server Edition environments that use single standby or multiple standby configurations.

Upgrading DB2 HADR Servers with One Standby (3 of 3)

- **STANDBY:** Upgrade instance: **db2iupgrade** <instancename>
 - db2iupgrade calls db2ckupgrade to verify that standby databases ready for upgrade.
 - **STANDBY:** **db2start**
 - **db2 UPGRADE DATABASE** <dbalias>
 - Upgrades metadata objects and if log validation succeeded in earlier steps, standby starts up and waits for a connection from primary.
 - Replay functionality begins in background and waits for upgrade log data to be received from primary.
 - Standby is considered "upgrade in progress". UPGRADE DB returns **SQL1103W**.
 - No user connections allowed while in "**upgrade in progress**" state.
 - Progress on standby can be monitored using db2pd -hadr in conjunction with DB2 diagnostics log.
 - Connect attempts can be issued, but receive SQL1776N, RC 9 as long as standby in upgrade in progress state.
 - **PRIMARY:** **db2 UPGRADE DATABASE** <dbalias>
 - Once primary database's metadata files upgraded, primary database looks to connect to standby within HADR timeout window.
 - Once connection formed, upgrade begins and log data is sent to standby for replay.
 - **PRIMARY:** Start using upgraded DB2 11.1 primary database:
 - **db2 ACTIVATE DATABASE** <dbalias>
- 74 Perform recommended post-upgrade tasks and verify upgrade successful. © 2016 IBM Corporation

When issue UPGRADE DATABASE command, it upgrades database's metadata files, like the database configuration, table space definitions, log header control files, and storage group files.

After upgrading the DB2 server, perform the recommended post-upgrade tasks such as resetting the diagnostic error level to its pre-upgrade value, adjusting log space size, and rebinding packages. In addition, verify that the upgrade of your DB2 server was successful.

Upgrading HADR Servers with Multiple Standbys Maintaining HADR role and without reinitializing HADR pair

Two Methods for Upgrading HADR Servers with Multiple Standbys:

1. METHOD 1: Upgrading all Multiple Standbys Together

- Upgrade all standby databases in parallel.



2. METHOD 2: Upgrading Multiple Standbys Separately

- Leave some auxiliary standby database at DB2 10.5 Fix Pack 7 or later until primary and principal standby have completed upgrade procedure in case of upgrade complications.



- Multiple standby upgrade procedure similar to single standby.

Only for single-partitioned DB2 databases, the upgrade of a HADR pair from a supported release and fixpack will no longer need the re-initialization of the standby through the use of a backup image shipped from the primary after upgrade. Re-initialization of the standby is still an option if the user wishes, but is no longer the recommended option.

The supported HADR upgrade procedure requires the primary and standby databases to both be, at a minimum, at DB2 10.5 Fix Pack 7 level. Any other prior level is not supported and attempts to use the documented procedure will fail. For these prior level databases, the standby will still have to be re-initialized by a backup image shipped from the primary after upgrade.

To help support the ability to roll forward (replay) through a database upgrade, two, new propagatable log records have been introduced. The database migration begin log record is written to mark the start of database upgrade, while the database migration end log record is written to mark the successful completion of database upgrade.

Method 1: Upgrading all HADR standbys together (1 of 5) Maintaining HADR role and without reinitializing HADR pair

Before You Begin:

- Review the system requirements for HADR: **db2pd -db <dbalias> -hadr**
- Both primary and standby database should be on same DB2 10.5 Fix Pack (7 or later).
- If have two HADR databases in same instance, perform role switch (if necessary) so that both primaries are on same system during upgrade: **db2 TAKEOVER HADR ON DB <dbalias>**
- Ensure you are familiar with steps involved in upgrading DB2 instance and DB2 databases.
- Ensure both primary's log shipping functionality and standby's log replaying functionality is working properly before running db2ckupgrade: **db2pd -db <dbalias> -hadr**
- If using Reads on Standby feature, ensure DB CFG **logindexbuild=on** so index recreation that is done during upgrade is sent to standby for replay.
 - Allows read connections to resume post upgrade on standby.
- In case of failures during HADR upgrade procedure, ensure familiar with "Dealing with failures while upgrading DB2 servers in HADR environments (10.5 Fix Pack 7 or later)".
- All DB2 upgrades, hardware upgrades, and software upgrades should be implemented in test environment before being applied to production system.

Upgrading your DB2 10.5 Fix Pack 7 or later databases to DB2 11.1 in a multiple standby environment is similar to the upgrade procedure for a single standby. This procedure maintains the database role and relies on normal log shipping and log replaying characteristics common to HADR functionality.

The procedure avoids the need to stop HADR for upgrade and avoids the need to reinitialize HADR. This reduces the window where no standby database exists and eliminates the cost of sending a backup image to all standby sites for re-initialization.

With multiple standby databases, you also have the added flexibility of upgrading all standby databases together to DB2 11.1 or leaving some auxiliary standby database at DB2 10.5 Fix Pack 7 or later until the primary and principal standby have completed the upgrade procedure in case of upgrade complications.

Method 1: Upgrading all HADR standbys together (2 of 5)

- **PRIMARY:** Monitor HADR databases. Reduces chance of upgrade failures.
 - Ensure primary log shipping and standby log replay are not lagging and function is healthy.
 - `db2pd -db <dbalias> -hadr` or `SELECT ... MON_GET_HADR ...`
- **STANDBYS:** Verify `HADR_REPLAY_DELAY=0`. Ensures standby log replay position can catch up to primary in reasonable amount of time.
- **PRIMARY:** Ensure log shipping is completed on primary and data is transferred to standbys by deactivating primary databases:
 - `db2 DEACTIVATE DATABASE <dbalias>`
- **PRIMARY:** Ensure applications do not connect so no new log data generated and standbys' log replay position eventually matches primary's log shipping position: `db2stop`
- **STANDBYS:** Ensure that all standby databases are active: `db2pd -db <dbalias> -hadr`
- **PRIMARY:** Upgrade DB2 instance: `db2iupgrade <instancename>`
 - `db2iupgrade` calls `db2ckupgrade` to verify primary database is ready for upgrade.
 - `db2iupgrade` does not run when `db2ckupgrade` reports errors. Check log file if encounter errors.
 - For HADR primary database from 10.5 Fix Pack 7 or later, `db2ckupgrade` verifies that valid standby database can be connected to primary database.
 - Once connection established, log shipping begins and ships any pending log data, if necessary.

The procedure depends on the DB2 10.5 Fix Pack 7 or later primary database having shipped all log data to the DB2 10.5 Fix Pack 7 or later standby databases and the standby databases having replayed all log data received. The DB2 11.1 standby database cannot replay log data from earlier DB2 versions. The procedure enforces this restriction through `db2ckupgrade` that is run during `db2iupgrade` or any `db2ckupgrade` invocation that establishes an exclusive connection to the database to do proper database consistency checking.

Method 1: Upgrading all HADR standbys together (3 of 5)

- Validates log positions between primary database and all standby databases. If log positions do not match for some database, then db2iupgrade/db2ckupgrade fails.
 - If fails, ensure log shipping/replay functionality is still healthy for that database.
 - If healthy, increase hadr_timeout for that database to give log validation check more time for log positions to match.
 - If some standby is unable to catch up to primary within enough time, remove that standby from hadr_target_list.
 - Removed standby database needs to be reinitialized post upgrade using a backup image.
 - For a given database, if there are issues getting log positions to match for principal and auxiliary standbys then must proceed by stopping HADR on that database. That database needs to be upgraded then HADR reinitialized.
- **STANDBYS:** Ensure log shipping is completed on primary databases and data is transferred to standbys. Deactivate all databases and stop the instance. Can do across all instances in parallel. **db2 DEACTIVATE DATABASE <dbalias>**
db2stop
- **STANDBYS:** Can upgrade across all instances in parallel. **db2iupgrade <instance>**
 - db2iupgrade calls db2ckupgrade to verify that standby databases ready for upgrade.

The procedure depends on the DB2 10.5 Fix Pack 7 or later primary database having shipped all log data to the DB2 10.5 Fix Pack 7 or later standby databases and the standby databases having replayed all log data received. The DB2 11.1 standby database cannot replay log data from earlier DB2 versions. The procedure enforces this restriction through db2ckupgrade that is run during db2iupgrade or any db2ckupgrade invocation that establishes an exclusive connection to the database to do proper database consistency checking.

This procedure can be applied to HADR databases in Enterprise Server Edition environments that use single standby or multiple standby configurations.

Method 1: Upgrading all HADR standbys together (4 of 5)

- **PRINCIPAL STANDBY:** **STARTING with the PRINCIPAL STANDBY FIRST:**
`db2start`
`db2 UPGRADE DATABASE <dbalias>`
 - Upgrades metadata objects and if log validation succeeded in earlier steps, principal standby starts up and waits for connection from primary.
 - Replay functionality begins in background and waits for upgrade log data to be received from primary.
 - Principal Standby is considered "upgrade in progress". UPGRADE DATABASE returns: `SQL1103W The UPGRADE DATABASE command was completed successfully.`
 - No user connections allowed while in "upgrade in progress" state.
 - Progress on standby can be monitored using `db2pd -hadr` in conjunction with DB2 diagnostics log.
 - Connect attempts can be issued, but receive `SQL1776N, RC 9` as long as standby in upgrade in progress state.
- **AUXILIARY STANDBYS:** Repeat above step for each auxiliary standby whenever convenient.
- **KEY:** Principal Standby be brought up first and sits waiting for primary **before** any auxiliary standby issues `UPGRADE DATABASE <dbalias>`

When issue UPGRADE DATABASE command, it upgrades database's metadata files, like the database configuration, table space definitions, log header control files, and storage group files.

Method 1: Upgrading all HADR standbys together (5 of 5)

- **PRIMARY:** For each database: **db2 UPGRADE DATABASE** <dbalias>
 - Upgrades primary database's metadata files
 - If log validation succeeded in earlier steps, primary database attempts connection with principal standby.
 - Upgrade will not process unless connection available to valid principal standby database.
 - Normal upgrade processing takes place.
 - All log data shipped to principal standby database for replay.
 - At any point once primary able to form connection with auxiliary database, log data shipped to auxiliary database for replay.
- **PRIMARY:** When upgrade completes, activate DB2 11.1 primary database:
db2 ACTIVATE DATABASE <dbalias>
- **STANDBYS:** When standby database has replayed all upgrade log data:
 - Standby is no longer considered in "upgrade in progress state" and stays activated.
 - Standby resumes normal operations.
 - Standby databases enabled for Reads on Standby take connections again.
- Perform recommended post-upgrade tasks and verify upgrade successful.

After upgrading the DB2 server, perform the recommended post-upgrade tasks such as resetting the diagnostic error level to its pre-upgrade value, adjusting log space size, and rebinding packages. In addition, verify that the upgrade of your DB2 server was successful.

Method 2: Upgrading HADR standbys separately (1 of 5)

Maintaining HADR role and without reinitializing HADR pair

Before You Begin:

- Review the system requirements for HADR: **db2pd -db <dbalias> -hadr**
- Both primary and standby database should be on same DB2 10.5 Fix Pack (7 or later).
- If have two HADR databases in same instance, perform role switch (if necessary) so that both primaries are on same system during upgrade: **db2 TAKEOVER HADR ON DB <dbalias>**
- Ensure you are familiar with steps involved in upgrading DB2 instance and DB2 databases.
- Ensure both primary's log shipping functionality and standby's log replaying functionality is working properly before running db2ckupgrade: **db2pd -db <dbalias> -hadr**
- If using Reads on Standby feature, ensure DB CFG **logindexbuild=on** so index recreation that is done during upgrade is sent to standby for replay.
 - Allows read connections to resume post upgrade on standby.
- In case of failures during HADR upgrade procedure, ensure familiar with "Dealing with failures while upgrading DB2 servers in HADR environments (10.5 Fix Pack 7 or later)".
- All DB2 upgrades, hardware upgrades, and software upgrades should be implemented in test environment before being applied to production system.

Upgrading your DB2 10.5 Fix Pack 7 or later databases to DB2 11.1 in a multiple standby environment is similar to the upgrade procedure for a single standby. This procedure maintains the database role and relies on normal log shipping and log replaying characteristics common to HADR functionality.

The procedure avoids the need to stop HADR for upgrade and avoids the need to reinitialize HADR. This reduces the window where no standby database exists and eliminates the cost of sending a backup image to all standby sites for re-initialization.

With multiple standby databases, you also have the added flexibility of upgrading all standby databases together to DB2 11.1 or leaving some auxiliary standby database at DB2 10.5 Fix Pack 7 or later until the primary and principal standby have completed the upgrade procedure in case of upgrade complications.

Method 2: Upgrading HADR standbys separately (2 of 5)

Leaving some auxiliary standby database at DB2 10.5 Fix Pack 7 or later until primary and principal standby have completed upgrade procedure.

- **PRIMARY:** Monitor HADR databases. Reduces chance of upgrade failures.
 - Ensure primary log shipping and standby log replay are not lagging and function is healthy.
 - `db2pd -db <dbalias> -hadr` or `SELECT ... MON_GET_HADR ...`
- **STANDBYS:** Verify `HADR_REPLAY_DELAY=0` on all standbys. Ensures standby log replay position can catch up to primary in reasonable amount of time.
- **PRIMARY:** Ensure log shipping is completed on primary and data is transferred to standbys by deactivating primary databases:
 - `db2 DEACTIVATE DATABASE <dbalias>`
- **PRIMARY:** Ensure applications do not connect so no new log data generated and standbys' log replay position eventually matches primary's log shipping position: `db2stop`
- **PRINCIPAL STANDBY:** Ensure all standbys are active: `db2pd -db <dbalias> -hadr`
- **PRIMARY:** Upgrade DB2 instance: `db2iupgrade <instancename>`
 - `db2iupgrade` calls `db2ckupgrade` to verify primary database is ready for upgrade.
 - `db2iupgrade` does not run when `db2ckupgrade` reports errors. Check log file if encounter errors.
 - For HADR primary database from 10.5 Fix Pack 7 or later, `db2ckupgrade` verifies that valid standby database can be connected to primary database.
 - Once connection established, log shipping begins and ships any pending log data, if necessary.

The procedure depends on the DB2 10.5 Fix Pack 7 or later primary database having shipped all log data to the DB2 10.5 Fix Pack 7 or later standby databases and the standby databases having replayed all log data received. The DB2 11.1 standby database cannot replay log data from earlier DB2 versions. The procedure enforces this restriction through `db2ckupgrade` that is run during `db2iupgrade` or any `db2ckupgrade` invocation that establishes an exclusive connection to the database to do proper database consistency checking.

Method 2: Upgrading all HADR standbys separately (3 of 5)

- Validates log positions between primary database and all standby databases. If log positions do not match for some database, then db2iupgrade/db2ckupgrade fails.
 - If fails, ensure log shipping/replay functionality is still healthy for that database.
 - If healthy, increase hadr_timeout for that database to give log validation check more time for log positions to match.
 - If some standby is unable to catch up to primary within enough time, remove that standby from hadr_target_list.
 - Removed standby database needs to be reinitialized post upgrade using a backup image.
 - For a given database, if there are issues getting log positions to match for principal and auxiliary standbys then must proceed by stopping HADR on that database. That database needs to be upgraded then HADR reinitialized.
- **STANDBYS:** Ensure log shipping is completed on primary databases and data is transferred to standbys. Deactivate all databases and stop the instance. Can do across all instances in parallel. **db2 DEACTIVATE DATABASE <dbalias> ; db2stop**
- **PRINCIPAL STANDBY: (STEP A:)**
 - db2start**
 - db2iupgrade <instance>**
 - db2iupgrade calls db2ckupgrade to verify that standby databases ready for upgrade.

The procedure depends on the DB2 10.5 Fix Pack 7 or later primary database having shipped all log data to the DB2 10.5 Fix Pack 7 or later standby databases and the standby databases having replayed all log data received. The DB2 11.1 standby database cannot replay log data from earlier DB2 versions. The procedure enforces this restriction through db2ckupgrade that is run during db2iupgrade or any db2ckupgrade invocation that establishes an exclusive connection to the database to do proper database consistency checking.

This procedure can be applied to HADR databases in Enterprise Server Edition environments that use single standby or multiple standby configurations.

Method 2: Upgrading all HADR standbys separately (4 of 5)

- **PRINCIPAL STANDBY: (STEP B:) FOR EACH DATABASE IN PRINCIPAL STANDBY INSTANCE:**

db2start

db2 UPGRADE DATABASE <dbalias>

- Upgrades metadata objects and if log validation succeeded in earlier steps, principal standby starts up and waits for connection from primary.
- Replay functionality begins in background and waits for upgrade log data to be received from primary.
- Principal Standby is considered "upgrade in progress". UPGRADE DATABASE returns: `SQL1103W The UPGRADE DATABASE command was completed successfully.`
- No user connections allowed while in "upgrade in progress" state.
- Progress on standby can be monitored using `db2pd -hadr` in conjunction with DB2 diagnostics log.
- Connect attempts can be issued, but receive `SQL1776N, RC 9` as long as standby in upgrade in progress state.

When issue UPGRADE DATABASE command, it upgrades database's metadata files, like the database configuration, table space definitions, log header control files, and storage group files.

Method 2: Upgrading all HADR standbys separately (5 of 5)

- **PRIMARY: db2 UPGRADE DATABASE** <dbalias>
 - Upgrade each primary database's metadata files in the primary instance
 - If log validation succeeded in earlier steps, primary database attempts connection with principal standby.
 - Upgrade will not process unless connection available to valid principal standby database.
 - Normal upgrade processing takes place.
 - All log data shipped to principal standby database for replay.
- **PRIMARY:** When upgrade on primary database completes, activate DB2 11.1:
db2 ACTIVATE DATABASE <dbalias>
- **PRINCIPAL STANDBY: (STEP X:)** When principal has replayed all upgrade log data:
 - Standby is no longer considered in "upgrade in progress state" and stays activated.
 - Standby resumes normal operations.
 - Standby databases enabled for Reads on Standby take connections again.
- **PRIMARY** and **PRINCIPAL STANDBY: (STEP Y:)** Confirm that everything is functioning as expected.
- **AUXILLARY STANDBYS:** Repeat **STEP A**, **STEP B**, **STEP X** and **STEP Y** for each to upgrade to DB2 11.1.
- **AUXILLARY STANDBYS:** Verify auxiliary standbys are healthy.
- Perform recommended post-upgrade tasks.

After upgrading the DB2 server, perform the recommended post-upgrade tasks such as resetting the diagnostic error level to its pre-upgrade value, adjusting log space size, and rebinding packages. In addition, verify that the upgrade of your DB2 server was successful.

Dealing with failure while upgrading HADR servers Maintaining HADR role & without reinit HADR pairs (1 of 5)

- **Scenario 1:** In single partition DB2 10.5 Fix Pack 7 or later, if primary's log shipping functionality and standby's log replay functionality are not healthy causing db2iupgrade/db2ckupgrade to fail.
 - If issue cannot be fixed within upgrade window, follow the HADR procedure that requires the stopping of HADR and reinitialization.
- **Scenario 2:** In single partition DB2 10.5 Fix Pack 7 or later, if primary's log shipping functionality and standby's log replay functionality are healthy but standby's replay position is still behind primary's log shipping position causing db2iupgrade/db2ckupgrade to fail.
 - Ensure that HADR_REPLAY_DELAY=0. Try to allow more time for standby to catch up by increasing hadr_timeout value. If neither of these options allow for log positions to match within upgrade window, then follow previous HADR procedure that requires the stopping of HADR and re-initialization.

The previous procedures maintained the HADR role and did not require the HADR pairs to be reinitialized. This procedure maintains the database role and relies on normal log shipping and log replaying characteristics common to HADR functionality. The procedure avoids the need to stop HADR for upgrade and avoids the need to reinitialize HADR. This reduces the window where no standby database exists and eliminates the cost of sending a backup image to the standby site for re-initialization.

During upgrade a failure can occur at any point in the procedure with any component that makes up the primary or standby. An upgrade is a scheduled event, so any failure is considered severe and having the primary or standby database available as quickly as possible is paramount. In most cases, a failure results in either the primary or standby database no longer being available to continue its role in the HADR upgrade procedure. When this happens, the failing database must be taken out of its role by stopping HADR, continuing upgrade as a non-HADR database, and then reinitializing HADR post upgrade.

It is difficult to document every possible failure scenario, but this topic attempts to walk you through what actions can be taken for failures at certain common points in the procedure.

Dealing with failure while upgrading HADR servers Maintaining HADR role & without reinit HADR pairs (2 of 5)

- **Scenario 3:** In single partition DB2 10.5 Fix Pack 7 or later, if primary database becomes unavailable preventing db2iupgrade/db2ckupgrade from being run.
 - If primary database cannot be brought back up within upgrade window, switch roles on standby and follow previous HADR procedure that requires stopping of HADR and re-initialization.
- **Scenario 4:** In single partition DB2 10.5 Fix Pack 7 or later, if standby database becomes unavailable preventing db2iupgrade/db2ckupgrade from being run.
 - If the standby database cannot be brought back up within upgrade window, follow previous HADR procedure that requires stopping of HADR and re-initialization.

The previous procedures maintained the HADR role and did not require the HADR pairs to be reinitialized. This procedure maintains the database role and relies on normal log shipping and log replaying characteristics common to HADR functionality. The procedure avoids the need to stop HADR for upgrade and avoids the need to reinitialize HADR. This reduces the window where no standby database exists and eliminates the cost of sending a backup image to the standby site for re-initialization.

During upgrade a failure can occur at any point in the procedure with any component that makes up the primary or standby. An upgrade is a scheduled event, so any failure is considered severe and having the primary or standby database available as quickly as possible is paramount. In most cases, a failure results in either the primary or standby database no longer being available to continue its role in the HADR upgrade procedure. When this happens, the failing database must be taken out of its role by stopping HADR, continuing upgrade as a non-HADR database, and then reinitializing HADR post upgrade.

It is difficult to document every possible failure scenario, but this topic attempts to walk you through what actions can be taken for failures at certain common points in the procedure.

Dealing with failure while upgrading HADR servers Maintaining HADR role & without reinit HADR pairs (3 of 5)

- **Scenario 5:** In DB2 11.1, if primary database becomes unavailable preventing upgrade procedure from continuing on standby.
 - If primary database cannot be brought back up within upgrade window, on standby issue STOP HADR followed by ROLLFORWARD DATABASE with STOP option. This will turn database into non-HADR database. Database will now be upgrade pending and so issue UPGRADE DATABASE to continue upgrade. Once complete conduct post testing. HADR must be reinitialized.
- **Scenario 6:** In DB2 11.1, if standby database becomes unavailable preventing UPGRADE DATABASE from starting up on primary.
 - If standby database cannot be brought back up within upgrade window, on primary issue STOP HADR. This turns database into non-HADR database. Database will still be upgrade pending so reissue UPGRADE DATABASE to continue upgrade. HADR will have to be reinitialized.

The previous procedures maintained the HADR role and did not require the HADR pairs to be reinitialized. This procedure maintains the database role and relies on normal log shipping and log replaying characteristics common to HADR functionality. The procedure avoids the need to stop HADR for upgrade and avoids the need to reinitialize HADR. This reduces the window where no standby database exists and eliminates the cost of sending a backup image to the standby site for re-initialization.

During upgrade a failure can occur at any point in the procedure with any component that makes up the primary or standby. An upgrade is a scheduled event, so any failure is considered severe and having the primary or standby database available as quickly as possible is paramount. In most cases, a failure results in either the primary or standby database no longer being available to continue its role in the HADR upgrade procedure. When this happens, the failing database must be taken out of its role by stopping HADR, continuing upgrade as a non-HADR database, and then reinitializing HADR post upgrade.

It is difficult to document every possible failure scenario, but this topic attempts to walk you through what actions can be taken for failures at certain common points in the procedure.

Dealing with failure while upgrading HADR servers Maintaining HADR role & without reinit HADR pairs (4 of 5)

- **Scenario 7:** In DB2 11.1, if standby database becomes unavailable while in upgrade in progress state.
 - Once UPGRADE DATABASE is issued on primary and primary forms connection with standby database, upgrade proceeds without issue on primary and eventually complete successfully. Concern is there is no standby database replaying log data, which leaves exposure to loss of primary. Post upgrade primary database can still be brought up through START HADR command specifying BY FORCE option. At this point, all attempts should be made to resolve issues with standby. Once resolved, since standby was in upgrade in progress state, UPGRADE DATABASE should be issued. Standby continues to replay upgrade log data shipped by primary until it completes and is no longer in upgrade in progress state.

The previous procedures maintained the HADR role and did not require the HADR pairs to be reinitialized. This procedure maintains the database role and relies on normal log shipping and log replaying characteristics common to HADR functionality. The procedure avoids the need to stop HADR for upgrade and avoids the need to reinitialize HADR. This reduces the window where no standby database exists and eliminates the cost of sending a backup image to the standby site for re-initialization.

During upgrade a failure can occur at any point in the procedure with any component that makes up the primary or standby. An upgrade is a scheduled event, so any failure is considered severe and having the primary or standby database available as quickly as possible is paramount. In most cases, a failure results in either the primary or standby database no longer being available to continue its role in the HADR upgrade procedure. When this happens, the failing database must be taken out of its role by stopping HADR, continuing upgrade as a non-HADR database, and then reinitializing HADR post upgrade.

It is difficult to document every possible failure scenario, but this topic attempts to walk you through what actions can be taken for failures at certain common points in the procedure.

Dealing with failure while upgrading HADR servers Maintaining HADR role & without reinit HADR pairs (5 of 5)

- **Scenario 8:** In DB2 11.1, if UPGRADE DATABASE with REBINDALL option was specified on primary and standby database becomes unavailable while in upgrade in progress state.
 - Difference from Scenario 7 is that on primary UPGRADE DATABASE command was specified with REBINDALL option. In this case, the UPGRADE DATABASE command requires and attempts a new connection to the database. If the standby database is not available during this second connection attempt, the UPGRADE DATABASE command returns SQL1499W. SQL1499W can be returned for many other reasons so the DB2 diagnostics log may be required to tell what failed and whether this scenario applies. If so, the primary database can still be brought up through the START HADR command specifying the BY FORCE option. Rebinding can still take place manually at this point. But, all attempts should be made to resolve the issues with the standby. Once resolved, since the standby was in upgrade in progress state, the UPGRADE DATABASE command should be issued. Standby continues to replay the upgrade log data shipped by the primary until it completes and is no longer in the upgrade in progress state.



The previous procedures maintained the HADR role and did not require the HADR pairs to be reinitialized. This procedure maintains the database role and relies on normal log shipping and log replaying characteristics common to HADR functionality. The procedure avoids the need to stop HADR for upgrade and avoids the need to reinitialize HADR. This reduces the window where no standby database exists and eliminates the cost of sending a backup image to the standby site for re-initialization.

During upgrade a failure can occur at any point in the procedure with any component that makes up the primary or standby. An upgrade is a scheduled event, so any failure is considered severe and having the primary or standby database available as quickly as possible is paramount. In most cases, a failure results in either the primary or standby database no longer being available to continue its role in the HADR upgrade procedure. When this happens, the failing database must be taken out of its role by stopping HADR, continuing upgrade as a non-HADR database, and then reinitializing HADR post upgrade.

It is difficult to document every possible failure scenario, but this topic attempts to walk you through what actions can be taken for failures at certain common points in the procedure.

Upgrading Automated Primary and Standby Database Maintaining HADR role and without reinitializing HADR pair

1. Ensure snapshot of resource model captured via **Issam** command.
Optionally, capture resource and policy info with **Issamctrl** or **sampolicy**.

TSA ↓
Resources

2. Delete cluster domain using **db2haicu** command. Deleting cluster domain is an irreversible so make sure have snapshot of resource model.
db2haicu -delete
3. Follow the procedure for "**Upgrading DB2 HADR Servers with One or Multiple Standbys; Maintaining HADR role and without reinitializing HADR pair**".
4. After upgrade completes, re-create resource model using resource model from **Issam** command.

Deleting the cluster domain is an irreversible step. Ensure that a snapshot of the resource model is captured via the **Issam** command before deleting the cluster domain. The resource model will need to be recreated once the DB2 upgrade has completed.

Upgrading DB2 servers in an automated HADR environment:

https://www.ibm.com/developerworks/community/blogs/IMSupport/entry/75_ways_to_demystify_db2_31_upgrading_db2_servers_in_an_automated_hadr_environment?lang=en_us&hootPostID=259ffd5591284888fab1b8a481941758

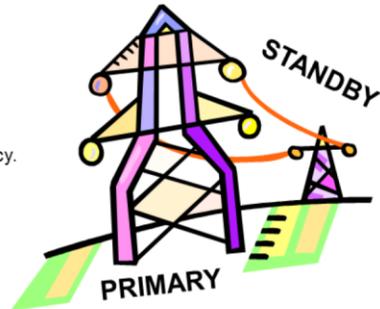
In an automated High Availability Disaster Recovery (HADR) environment, rolling upgrade procedure will not work when upgrading from an earlier to a later version of a DB2 database system.

Upgrading DB2 servers with automated HADR and WITH Re-initialization of Standby

1. As instance owner, stop HADR on both standby and primary servers:
`db2 stop hadr on database sample`
2. Verify database role has changed to standard on both servers using:
`db2 get db cfg for sample | grep -i role`
3. Ensure snapshot of resource model captured via `Issam` command.
Optionally, capture resource and policy info with `Issamctrl` and `sampolicy`.

TSA Resources

4. Delete cluster domain using `db2haicu -delete`
5. Deactivate database on both standby and primary servers:
`db2 deactivate db sample`
6. Stop DB2 instance: `db2stop`
7. As root, install DB2 11.1 in new directory path.
8. As root, upgrade DB2 instance: `DB2DIR/instance/db2iupgrade db2inst1`
Where `DB2DIR` represents installation location of new release of DB2 database system installed.
9. As instance owner, start DB2 instance: `db2start`
10. As instance owner, upgrade database:
`db2 upgrade database sample`
11. Reinitialize HADR by following steps documented in Knowledge Center, "Initializing high availability disaster recovery (HADR)".
12. Once HADR pair is reestablished, recreate cluster domain resources. See "Automated HADR configuration setup" whitepaper: <http://www.ibm.com/developerworks/data/library/long/dm-0907hadrdb2haicu/>



In an automated High Availability Disaster Recovery (HADR) environment, rolling upgrade procedure will not work when upgrading from an earlier to a later version of a DB2 database system.

Note : Deleting the cluster domain is an irreversible step. Ensure that a snapshot of the resource model is captured via the `Issam` command before deleting the cluster domain. The resource model will need to be recreated once the DB2 upgrade has completed.

Upgrading DB2 servers in an automated HADR environment:

https://www.ibm.com/developerworks/community/blogs/IMSupport/entry/75_ways_to_demystify_db2_31_upgrading_db2_servers_in_an_automated_hadr_environment?lang=en_us&hootPostID=259ffd5591284888fab1b8a481941758

Post-upgrade tasks for DB2 servers (1 of 2)

- If reset diaglevel to 3 or higher as recommended in pre-upgrade tasks, reset to value before upgrade.
- Existing pre-DB2 11.1 tables with row compression enabled have classic row compression enabled. If want to use adaptive compression, alter tables.
- If changed log space setting as recommended in pre-upgrade tasks, reset logfilesiz, logprimary, and logsecond to pre-upgrade values.
- If changed logindexbuild, reset to value before upgrade.
- Ensure existing libraries for external routines remain on original location before upgrade.
- Activate database to start database and all necessary database services.
- Automatic storage table spaces inherit media attribute values, including overhead, device read rate and data tag attributes, from the storage group it is using by default. After upgrading to DB2 11.1, existing table spaces retain their settings and OVERHEAD and DEVICE READ RATE attributes for the storage group are set to undefined. Can set media attributes with the ALTER STOGROUP statement. [↗](#)
- Consider setting new registry variables and configuration parameters that can impact behavior of DB2.
- If automatic collection of statistics failed on certain system catalog tables during database upgrade, update statistics on these tables.
- If did not use REBINDALL on UPGRADE DATABASE then rebind packages to validate packages and use updated statistics or new index information.

After you upgrade the DB2 server, perform the recommended Postupgrade tasks for DB2 servers such as resetting the diagnostic error level, adjusting log space size, and rebinding packages. In addition, verify that the upgrade of your DB2 server was successful.

Perform the above post-upgrade tasks that apply to your DB2 server.

Post-upgrade tasks for DB2 servers (2 of 2)

- Refresh data in existing MQTs by using REFRESH TABLE. MQTs on unicode databases using language aware collation and involves LIKE predicate or substring function involved in a basic predicate, need to be refreshed.
- Migrate Explain Tables to retain explain information previously gathered.
- If obtained customized code page conversion tables from DB2 support service, copy all files for those tables from the *DB2OLD/conv* to *DB2DIR/conv*. Do not have to copy standard code page conversion tables.
- If upgraded existing DB2 10.5, 10.1 or 9.7 on Windows, can restore customized code page conversion tables that backed up as part of pre-upgrade tasks.
- Upgrade existing target tables for event monitors that write to tables and to unformatted event (UE) tables by using new EVMON_UPGRADE_TABLES procedure.
- Test applications and tools to ensure DB2 server is working as expected.
- Back up your databases after DB2 server upgrade is complete.
- Previous log files are now treated like log files from current release and regular DB2 log file management will occur. If upgraded from DB2 9.7, no log archiving will occur for log files in the original active log paths.
- Must migrate SQL Replication in order to support new LSN formats.

93

© 2016 IBM Corporation

After upgrading DB2 servers, you should perform several post-upgrade tasks to ensure that your DB2 servers perform as expected and at their optimum level.

Perform the above post-upgrade tasks that apply to your DB2 server.

For recoverable databases, DB2 no longer renames log files in the active log paths to the .MIG extension. Log files from previous releases are now treated like log files from the current release and regular DB2 log file management will occur. If you upgraded from DB2 Version 9.7, no log archiving will occur for log files in the original active log paths. You will be responsible for the management of these log files. The db2cklog utility can be used to determine what log files in the active log paths are from the pre-DB2 Version 11.1 release. If necessary, this can be used to assist in manual log file management.

If you obtained customized code page conversion tables from the DB2 support service, you need to backup all of the files in the DB2OLD\conv directory where DB2OLD is the location of your existing pre-11.1 DB2 copy. You do not need to backup standard code page conversion tables. Upgrading your pre-11.1 DB2 copy removes these tables because standard code page tables are contained in a DB2 11.1 library.

Post upgrade - What to do next?

- If upgraded existing DB2 10.5, 10.1, or 9.7 copy, database log directories have changed. Review db2diag.log file which has entries detailing new log directories. If user defined log directory /usr/logpath is used, after upgrade the location of log files will be /usr/logpath/NODE0000/LOGSTREAM0000.
- If upgrade DB2 server running HADR, and during upgrade to DB2 11.1, database role was changed from primary to standard, then reinitialize HADR.
- When DB2 server performance is stable, take advantage of optimizer improvements and collect statistics for new functionality by updating statistics for your upgraded databases. During database upgrade to DB2 11.1, statistics collected from existing database tables retain their values. Statistics for new characteristics on tables and indexes have a value of -1 to indicate there is no information gathered. Only need these statistics if using new functionality.
- After updating statistics for upgraded databases, determine if index or table reorganization is necessary by running REORGCHK. Table and index reorganization can help improve performance.
- At this point, resume all of maintenance activities such as backing up databases and updating statistics. Remove any DB2 10.5, 10.1, 9.7 or 9.8 copies that are no longer need.

Perform the above post-upgrade tasks that apply to your DB2 database products or add-on features.

Adjusting log space size if upgrade from DB2 9.7

- Set appropriate size for log files since is one of important factors in tuning server.
- If increased log files sizes as pre-upgrade task, can restore additional free space to server.
- On partitioned database, must adjust log space size on catalog database partition.
- Restore log file size settings to values before upgrade:
db2 UPDATE DB CFG FOR sample using LOGSECOND *previous-value*
- If you enabled infinite active logging, disable it by running the following commands:
db2 UPDATE DB CFG FOR sample using LOGARCHMETH1 *previous-value*
db2 UPDATE DB CFG FOR sample using LOGSECOND *previous-value*
- To support larger log record headers, increase log space setting, by approximately 10% - 15% over what used for DB2 9.7.
- To support larger log record headers, increase logbufsz and softmax by 10% - 15% over what used for DB2 9.7.
- logfilsiz changes take effect only when database is reactivated.



Adjust the log space size in upgraded databases as needed.

Download DB2 Bind files – sqllib/bnd/*.bnd

- Included in DB2 install image are bind files (sqllib/bnd/*.bnd) containing packages for various database and CLI utilities.
 - When remote client executes utilities, it's necessary for package associated with remote client's version and release be bound to database first.
 - Often difficult to track down client's image to bind .bnd files.
 - Application may execute on DB2 Runtime client of different version than DB2 11.1 server. If client attempts to execute utility (such as CLP, Import, Export, ...) may generate SQL0805N because package associated with version of utility not bound to new DB2 11.1 database.
 - Bind packages using bind files from client version.
- Download tar of pre-11.1 client bind files from following website:



<http://www-01.ibm.com/support/docview.wss?uid=swg21501302&myns=swgimgmt&mynp=OCSSSNY3&mynp=OCSSEPGG&mynp=OCSSEPDU&mync=E>

Included in every DB2 install image are bind files (sqllib/bnd/*.bnd) containing the packages for various database and CLI utilities. Many of these bind files correspond to utilities that reside in the Runtime client libraries. When a remote client executes one of these utilities, it is necessary for the package associated with that remote client's version and release to be bound to the database first.

When creating a new database that may accept connections from various remote clients which may run these utilities, it is often difficult to track down each client's image to bind the .bnd files. If you deploy a new database created on DB2 11.1, an application that is executing on a remotely connected client may be running a DB2 Runtime client of a different version. If the application attempts to execute a utility (such as CLP, Import, Export, etc...) it could generate an SQL0805N because the package associated with this version of the utility was not bound to the new DB2 11.1 database. You will have to bind the packages using the bind files from client version. You can download the tar file from the following website and use the bind files from it.

<http://www-01.ibm.com/support/docview.wss?uid=swg21501302&myns=swgimgmt&mynp=OCSSSNY3&mynp=OCSSEPGG&mynp=OCSSEPDU&mync=E>

How to revalidate invalid DB2 SYSTEM views after upgrade?

After update run query to see if invalid system views after upgrade:

```
db2 "select name, valid from sysibm.sysviews"
```

<u>NAME</u>	<u>VALID</u>
SQLFOREIGNKEYS	N

Revalidate with administrative routine.

```
db2 "CALL SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS  
('view', 'MY_SCHEMA', 'MY_VIEW')"
```

The ADMIN_REVALIDATE_DB_OBJECTS procedure revalidates database objects.

This procedure takes various input parameters that control the level of revalidation that is performed:

- To revalidate all of the invalid objects in the database, either specify NULL for all parameters, or call the procedure without parameters.
- To revalidate all of the invalid database objects under a specific schema, specify a value for object_schema, and specify NULL for object_name and object_type.
- To revalidate a specific invalid database object, specify valid values for the first parameters.
- To force the revalidation of compiled SQL PL or PL/SQL objects regardless of their state being valid or invalid, specify a value for object_type, object_schema, object_name, and specify 'Y' for force.
- To enable or disable debug on compiled SQL PL or PL/SQL objects being revalidated, specify 'Y' or 'N' for debug_flag.

This procedure will revalidate only invalid objects and regular or materialized query tables in reorg-pending state. All invalid objects can be found in SYSCAT.INVALIDOBJECTS. To find out which tables are in reorg-pending state, use the ADMIN_GET_TAB_INFO table function.

Upgrade database - runstats on system tables

- UPGRADE DATABASE automatically collects statistics for all system catalog tables during upgrade.
- **If manually modified SYSTATS views, reissue updates.**
- *Which RUNSTATS is called for automatic statistics collection?*

Auto_runstats	User Profile	RUNSTATS Command
Enabled	Exists	RUNSTATS command with SET PROFILE parameter using information in STATISTICS_PROFILE column in SYSCAT.TABLES
Enabled	Does not Exist	RUNSTATS command with default parameters
Disabled	N/A	auto_runstats 1

[1](#) If statistics previously collected for table, RUNSTATS is issued as indicated in table. If there are no statistics collected for table, RUNSTATS is not issued.

The UPGRADE DATABASE command automatically collects statistics for all system catalog tables during database upgrade.

If you have manually modified your system catalog table statistics via updates to SYSSTATS views, manually reissue these updates to the SYSSTATS views.

Runstats on objects to use new functionality

- **Runstats** on objects so optimizer uses new functionality
 - DB CFG **AUTORUNSTATS=ON**
 - **db2 RUNSTATS ON TABLE mel.t1 WITH DISTRIBUTION AND DETAILED INDEXES ALL**
 - or update statistics on multiple objects:
db2 REORGCHK UPDATE STATISTICS ON TABLE [ALL | SCHEMA <schema-name> | SYSTEM]
- 
- Complete Runstats first before rebind static embedded SQL packages!
 - Then rebind all valid and invalid packages in each database.
 - **Could specify REBINDALL on UPGRADE DATABASE but before runstats on objects.**
 - **db2rbind database-name -l logfile all**
 - From IBM Data Studio, can open task assistant to rebind packages.
 - Applies only to C, C++, COBOL, FORTRAN, REXX embedded static SQL applications.
 - Can request access plans for static SQL be preserved across binds or rebinds because don't want existing access plan to change.
ALTER PACKAGE/BIND/REBIND/PRECOMPILE ... APREUSE YES
SYSCAT.PACKAGES.APREUSE = Y when completed.
 - Test your applications and tools to ensure that the server is working as expected.
 - After rebound all packages, can automatically take advantage of optimizer enhancements.

Runstats on objects so optimizer uses new functionality. You can use reorgchk to update statistics on multiple objects.

During database upgrade, all packages for user applications and routines are marked as invalid. Packages implicitly rebound first time used after upgrade database. To eliminate overhead, explicitly rebind invalid packages. Must explicitly rebind inoperative packages. Alternatively, specify REBINDALL on UPGRADE DATABASE. Applies only to C, C++, COBOL, FORTRAN, and REXX embedded SQL applications.

The REBINDALL option specifies that a REBIND of all packages is performed during upgrade. Performing the REBINDs automatically during the upgrade ensures that the step is not missed and will help ensure that other applications are not started before the REBINDs are complete.

Reference: Student Notes

When a new DB2 data server release is installed, existing packages must be rebound in order for access sections to be rebuilt, so that they are compatible with runtime component of new data server release. However, you may not want the existing access plan to change. You can enable access plan reuse through ALTER PACKAGE statement or by using the APREUSE option on BIND, REBIND or PRECOMPILE. Packages that are subject to access plan reuse have the value "y" in APREUSE columns of the SYSCAT.PACKAGES catalog view.

Upgrade explain tables & rerun benchmarks

- To maintain explain table info gathered from previous releases, upgrade explain tables to DB2 11.1.

db2exmig -d dbname -e explain_schema

OR

CALL SYSPROC.SYSINSTALLOBJECTS ('EXPLAIN', 'M', CAST (NULL AS VARCHAR(128)), CAST (NULL AS VARCHAR(128)))

- Renames original explain tables, creates new set of tables by using EXPLAIN.DDL file, and copies contents of original explain tables to new tables. Finally, tool drops original explain tables and preserves any user-added columns in explain tables.
- After upgrade database, update statistics, rerun benchmarks, compare and examine explain output if necessary.
 - Use **db2batch** benchmark tool to collect elapsed and CPU times
db2batch -d DBX -f input.sql -r output.fil -i complete -o e yes -isol CS
 - Keep **db2exfmt** output for each test query

101

© 2016 IBM Corporation

Run a number of performance tests before upgrading your DB2 server. The **db2batch** benchmark tool helps you to collect elapsed and CPU times for running queries. You can use this tool to develop performance tests. Record the exact environment conditions where you run your tests.

Also, keep a record of the **db2exfmt** command output for each test query. Compare the results before and after upgrade. This practice can help to identify and correct any performance degradation that might occur.

The SYSINSTALLOBJECTS procedure can migrate explain tables. The M option is only valid when used with the tool name EXPLAIN. This option migrates explain tables to be compatible with the current version.

The db2batch tool reads SQL statements and XQuery statements from a flat file, dynamically prepares and describes the statements, returns an answer set, populates explain tables, and measure the elapsed time to run each statement where the prepare, execute, and fetch times are reported separately. The default isolation level is RR.

The db2exfmt tool can be used to analyze explain output stored in explain tables.

Upgrade event monitor output tables

- As event monitors enhanced, tables require new columns.
- **EVMON_UPGRADE_TABLES** stored procedure
 - upgrades definitions of existing tables along with all data
 - automatically deactivate event monitor before upgrade
 - automatically reactivates after upgrade completed

call **EVMON_UPGRADE_TABLES** (*evmon_name*,
evmon_type, *options*, *num_evmons_evaluated*,
num_evmons_to_upgrade, *num_evmons_upgraded*)

or

call **evmon_upgrade_tables** (null, null, null, ?, ?, ?)

- Can also use ALTER EVENT MONITOR statement to add new logical groups to event monitor as alternative

You can upgrade event monitor output tables after you upgrade the DB2 product. This capability lets you retain any data that might exist in event monitor tables that you had before you upgraded.

As event monitors are enhanced in the DB2 product, the tables they produce might change. For example, new columns might be added to a table for reporting new monitor elements. Previously, if you had existing event monitors that wrote to tables that contained data that you wanted to retain, and you wanted to collect the data in the newly-added columns, you were required to manually alter them after upgrading to the new release. This alteration involved adding any of the new columns that you might want to use. If you did not add the new columns, the event monitor would work as it had in the previous release, capturing only the data supported by that the event monitor in that release. Unformatted event tables that had changed could not be upgraded at all; you were required to drop them and then re-create them. The EVMON_UPGRADE_TABLES stored procedure upgrades the definitions of existing event monitor tables to match those produced by the current level of the DB2 product. This feature lets you keep any existing tables that you might have, along with all the data they contain, eliminating the need to manually alter, or to drop and re-create tables. The EVMON_UPGRADE_TABLES procedure works with both regular and UE tables. For regular tables, the procedure adds any new columns needed, drops old columns that are no longer required, and alters any columns as needed. For UE tables, the procedure adds new columns and modifies existing columns as needed to allow the UE table to be processed by the db2evmonfmt tool, or the EVMON_FORMAT_UE_TO_TABLES or EVMON_FORMAT_UE_TO_XML routines.

Important: Any active event monitors must be deactivated for the upgrade process to work properly. The EVMON_UPGRADE_TABLES procedure automatically deactivates any active event monitors before it begins upgrading tables. Do not reactivate any event monitors with tables being processed by EVMON_UPGRADE_TABLES, or the upgrade process will fail. Any event monitors that were active before the upgrade are activated again after the upgrade has completed.

Implications of not upgrading event monitor tables

As in past releases, you can choose to not upgrade your event monitor tables. However, any new columns that have been added to the event monitor in the new release will not be populated with data, and will not be available for queries. Also, the values for any monitor elements that previously existed in the old release and that increased in size in the new release might be truncated. For example, if a monitor element increases in size from VARCHAR(20) to VARCHAR(128) in the new release, and you do not upgrade the previously-existing tables, the column that contains the monitor element values will continue to store only 20 characters of data, even though the system may be sending 128-bytes of data for that monitor element to the event monitor.

Back up databases before AND after upgrade

- Backup used for reversing instance/database upgrade.
- **REMEMBER: After upgrade instances to 11.1, cannot backup databases until upgrade them!**
- **Can optionally take ONLINE BACKUP if all of following are true:**
 - Updating database "in place".
 - DB2 Enterprise Server Edition single partition 10.5 Fix Pack 7 or pureScale 10.5 Fix Pack 7 or later.
 - Ensure have full database backup image and log files from pre-DB2 11.1 release and log files from 11.1 after upgrade completes.
 - Ability to roll-forward through database version upgrades
- **OFFLINE BACKUP required if any of the following are true:**
 - Restoring pre-DB2 11.1 backup into new DB2 11.1 instance
 - Multiple partitioned database. **Must back up all database partitions.**
 - Not DB2 or pureScale 10.5 Fix Pack 7 or later
 - If performed full online or offline database backup recently and cannot perform another one before upgrading, can perform incremental offline database backup instead if TRACKMOD=ON.
 - If activated and configured DB2 Advanced Copy Services (ACS), can use USE SNAPSHOT parameter to perform snapshot backup. Can only restore snapshot backup to instance of same version. Cannot use snapshot backup to upgrade to new version.



Before and after the upgrade process to DB2 11.1, it is strongly recommended that you perform a full database backup. If an error occurs during the upgrade process, you need full database backups to recover and upgrade your databases.

After you upgrade your instances to DB2 11.1, you cannot backup databases until you upgrade them.

Database upgrade no longer renames log files

- When upgrading from DB2 9.7, 10.1 or 10.5, database log files no longer renamed from .LOG to .MIG.
- For 10.1 and 10.5, any unarchived files prior to upgrade will be archived from uplevel and will not require any action.
- For 9.7, any unarchived files will not be archived, and will be left in old log directories. Users will have to back up, archive, and manage files manually.
- db2cklog tool assists in manual management of log files to confirm what logs are from previous DB2 versions. If recovery log file not from current version, then DBT warning message returned.



db2cklog CHECK 80 to 94 ARCHLOGPATH <path where logfiles stored>

- Default is current directory

Reversing Upgrade: Since no longer rename log files from previous DB2 versions, if need to reverse upgrade, before dropping database, log files in active log directories should be manually saved before proceeding.

Prior to 11.1, for recoverable databases, UPGRADE DATABASE renames all log files in active log path with the extension .MIG. After upgrade databases successfully, you deleted all S*.MIG files. In DB2 11.1, a database upgrade no longer renames log files.

Two new DB2 Log Records introduced - Recovering through DB2 server upgrade

To support ability to roll forward (replay) through database upgrade, two new propagatable log records have been introduced.

1. Database Migration Begin Log Record

- Written to mark start of database upgrade.

2. Database Migration End Log Record

- Written to mark successful completion of database upgrade.

▪ Supported for

- Single-partition database upgrading from DB2 10.5 Fix Pack 7 or later, and
- DB2 pureScale from DB2 10.5 Fix Pack 7



For single-partition DB2 users upgrading from DB2 Version 10.5 Fix Pack 7, or later, high availability disaster recovery (HADR) environments can now be upgraded without the need to re-initialize the standby database after performing an upgrade on the primary database.

DB2 pureScale users upgrading from DB2 10.5 (all Fix Packs) still required to stop HADR before upgrading and reinitialize standby post-upgrade.

Recovering through DB2 server upgrade (info 1)

- **Take new full online database backup in DB2 11.1. Once backup completes and verified, this procedure is not necessary!**
- Database upgrade is recoverable operation if have backup image and log files from pre-DB2 11.1 release and log files from 11.1. Then post upgrade, can restore and roll forward to a point in time before any failure.
- DB2 11.1 **can not** restore pre-11.1 online backup image.
- DB2 11.1 **can not** roll forward through pre-11.1 log files.
- Pre-11.1 backup image must be restored in DB2 10.5 Fix Pack 7 or later
- Requires reinstall previous release before restore & roll forward take place.
 - Roll forward must be issued from DB2 10.5 Fix Pack 7 or later.
- Roll forward replays all log records created in pre-DB2 11.1 until comes to end of release. Now you can choose to:
 - Stay in previous release, consider reversing or fall back of upgrade.
 - Move to DB2 11.1 and continue roll forward through upgrade.
- If intend to roll forward to point in time post upgrade in DB2 11.1:
 - Re-install DB2 11.1 and continue roll forward to point in time you want.
- 106 – Examine all post upgrade tasks.

For recoverable databases, as part of the upgrade procedure to DB2 11.1, you must devise a recovery plan in case issues arise during the upgrade or post upgrade. Database upgrade is a recoverable operation so if you have a backup image and log files from a pre-DB2 11.1 release and log files from DB2 11.1 then post upgrade you can restore and roll forward to a point in time before any failure.

If you intend to roll forward to a point in time post upgrade in DB2 11.1 then you must re-install DB2 11.1 and continue the roll forward to the point in time you want. Once the database comes out of roll forward pending state then all post upgrade tasks should be reexamined to make sure that the database is ready to accept new workload. For example, this could involve rebinding of packages. If the database configuration parameter logindexbuild was not on during the initial upgrade, then during roll forward, catalog indexes are marked bad. First connect recreates these indexes, which might create a delay before data can be accessed.

It is highly suggested that a new online database backup be created at your earliest convenience so that you do not have to rely on this recovery procedure again.

Recovering through a DB2 server upgrade (info 2)

- **Ensure have root access on Linux and UNIX operating systems or Local Administrator authority on Windows.**

- **Ensure have SYSADM authority.**

- **Perform following steps before upgrade DB2 server:**
 - Review upgrade recommendations and disk space requirements.
 - Before upgrade, ensure have valid pre-DB2 11.1 backup image and valid log files for all databases.
 - After upgrade, ensure have valid log files taken in DB2 11.1 for all databases.
 - Back up all database manager configuration parameter values for each instance.
 - Back up all database configuration parameter values for each database.
 - Perform other pre-upgrade tasks that apply to your environment.

For recoverable databases, as part of the upgrade procedure to DB2 11.1, you must devise a recovery plan in case issues arise during the upgrade or post upgrade.

Recovering Through DB2 Server Upgrade (info 3)

- **Ability to roll-forward through database version upgrades for single-partition databases and DB2 pureScale**
 - Applies only to DB2 server upgrade. Does not apply to DB2 client upgrade.
 - Not supported in partitioned database environments.
- Take an online full backup of all databases going to upgrade.
- Does not work if previous database version is DB2 9.7, 10.1 or 10.5 Fix Pack 6 or earlier.
- **Should only be used for failures that occur after upgrading to DB2 11.1 and before new full database online backup completes.**
 - Should only be used to recover a database that has successfully been upgraded to DB2 11.1 and where user workload transactions have been performed that would be lost otherwise.
 - Should only be used on major failure that can only be recovered restoring most recent backup, for example, storage failures or data corruptions.
- Also could be used in cold standby system to recover database after upgrade and before new full database online backup completes. **Cold standby system should remain at 10.5 Fix Pack 7 or later until new full database online backup completes on main system.** Recovery procedure on cold standby system starts with step that restores pre-DB2 11.1 backup.

108

© 2016 IBM Corporation

Recovery in a multiple database scenario when only a subset requires recovery:

In case only a subset of the databases require recovery the procedure below does not change. In this case, all databases will take an outage while the subset that needs recovery are recovered. The steps can be optimized by moving only once to the previous release then doing the recovery for all databases to the end of release, followed by moving up to DB2 11.1 once, and then finishing the recovery for all databases.

In the previous release, the restore can consist of rebuilding a database. It is highly recommended to choose all table spaces in the database. If choosing a subset of table spaces, be aware that once you continue the roll forward in DB2 11.1 any table space in restore or roll forward pending can not be recovered. Those table spaces will have to be dropped and re-created.

Recovering through DB2 single partition server upgrade on Linux and UNIX (detailed steps 1 of 3)

- **REMEMBER TO COMPLETE STEPS FOR EACH DATABASE!**
- **instance owner:** IMPORTANT to protect log files! Necessary as some or all of log files in active or mirror log paths may not be archived and drop of database deletes those log files. To get current value of log paths, use:
db2 GET DBM CFG | grep -i log
 - Note: Retain copy until recovery procedure completed and new online full database backup completed and verified, which includes copy of log files.
- **instance owner:** Record database path where database(s) restored and authentication:
db2 LIST DB DIRECTORY
- **instance owner:** Drop database(s) that need recovery in DB2 11.1:
db2 DROP DB <dbalias>
- **instance owner:** stop instance: **db2stop**
- **instance owner:** drop DB2 11.1 instance (does not remove DB files):
/opt/IBM/db2/copy1/instance/db2idrop <instancename>

Use this procedure to recover one or more databases through a DB2 single partition Linux or UNIX server upgrade.

Make sure that the backup image that is used to perform recovery is accessible and previously validated.

Ensure that you have root access on Linux and UNIX operating systems or Local Administrator authority on Windows. Ensure that you have SYSADM authority.

Recovering through DB2 single partition server upgrade on Linux and UNIX (detailed steps 2 of 3)

- **Root:** Re-create instance in pre-DB2 11.1:
pre-DB2 11.1_DB2DIR/instance/db2icrt -u db2fenc1 db2inst1
- **instance owner:** Restore database manager configuration parameter values:
db2 UPDATE DBM CFG USING <parameter> <value>
- **instance owner:** start instance: **db2start**
- **instance owner:** Restore database(s) from pre-DB2 11.1 backup into recorded database paths: **db2 RESTORE DB <dbalias> FROM ... DBPATH ...**
- **instance owner:** Roll forward database(s) to a point in time before failure.
Ensure log archive location available. Supply saved log files:
 - Copying log files back into active log paths or overflowlogpath
 - Supplying OVERFLOW LOG PATH parameter on ROLLFORWARD DB command.
db2 ROLLFORWARD DB <dbalias> TO <isotime> OVERFLOW LOG PATH <pathname>
 - When ROLLFORWARD returns **SQL2463N** or **SQL2464N**, means end of release is reached. Keeps database in "Rollforward Pending" condition.
 - **Do not use "AND STOP" clause on ROLLFORWARD command.**
- **instance owner:** Stop instance: **db2stop**
- **instance owner:** Uncatalog database(s): **db2 UNCATALOG DB <dbalias>**

110

© 2016 IBM Corporation

Perform the following steps before you upgrade your DB2 server:

- Review upgrade recommendations and disk space requirements.
- Before upgrade, ensure that you have a valid backup image and valid log files for all databases taken in the previous release.
- After upgrade, ensure that you have valid log files for all databases taken in DB2 11.1.
- Back up all database manager configuration parameter values for each instance and all database configuration parameter values for each database.
- Perform other pre-upgrade tasks that apply to your environment.

Recovering through DB2 single partition server upgrade on Linux and UNIX (detailed steps 3 of 3)

- **Root:** move prior release's sqllib directory.
- **Root:** upgrade instance to DB2 11.1:
DB2 11.1_DB2DIR/instance/db2iupgrade <instancename>
- **instance owner:** start instance: **db2start**
- **instance owner:** Re-catalog database(s) with recorded database path and authentication: **db2 CATALOG DB** <dbname> **AS** <dbalias> **ON** <database-path> **[AUTHENTICATION authentication-type]**
- **instance owner:** For database(s) that need recovery, continue roll forward until complete.
db2 ROLLFORWARD DB <dbalias> **TO** <isotime> **[AND STOP]**
- Once database comes out of roll forward pending state, examine all post upgrade tasks.

Use this procedure to recover one or more databases through a DB2 single partition Linux or UNIX server upgrade.

Recovering through DB2 single partition server upgrade on Windows (detailed steps 1 of 3)

- **REMEMBER TO COMPLETE STEPS FOR EACH DATABASE!**
- **instance owner:** IMPORTANT to protect log files! Necessary as some or all of log files in active or mirror log paths may not be archived and drop of database deletes those log files. To get current value of log paths, use:
db2 GET DB CFG FOR <dbalias> | find /i log
 - Note: Retain copy until recovery procedure completed and new online full database backup completed and verified, which includes copy of log files.
- **instance owner:** Record database path where database(s) restored and authentication:
db2 LIST DB DIRECTORY
- **instance owner:** Drop database(s) that need recovery in DB2 11.1:
db2 DROP DB <dbalias>
- **instance owner:** stop instance: **db2stop**
- **Administrator:** Uninstall your DB2 copy through Control Panel > Programs. This command does not remove the database files.
- **Administrator:** Re-create your DB2 copy: **db2setup**

Use this procedure to recover one or more databases through a DB2 single partition Linux or UNIX server upgrade.

Make sure that the backup image that is used to perform recovery is accessible and previously validated.

Ensure that you have root access on Linux and UNIX operating systems or Local Administrator authority on Windows. Ensure that you have SYSADM authority.

Recovering through DB2 single partition server upgrade on Windows (detailed steps 2 of 3)

- **instance owner:** Restore database manager configuration parameter values:
db2 UPDATE DBM CFG USING <parameter> <value>
- **instance owner:** start instance: **db2start**
- **instance owner:** Restore database(s) from pre-DB2 11.1 backup into recorded database paths: **db2 RESTORE DB <dbalias> FROM ... DBPATH ...**
- **instance owner:** Roll forward database(s) to a point in time before failure.
Ensure log archive location available. Supply saved log files:
 - Copying log files back into active log paths or overflowlogpath
 - Supplying OVERFLOW LOG PATH parameter on ROLLFORWARD DB command.
db2 ROLLFORWARD DB <dbalias> TO <isotime> OVERFLOW LOG PATH <pathname>
 - When ROLLFORWARD returns **SQL2463N** or **SQL2464N**, means end of release is reached. Keep database in "Rollforward Pending" condition.
 - Do not use "AND STOP" clause on ROLLFORWARD command.
- **instance owner:** Stop instance: **db2stop**
- **instance owner:** Uncatalog database(s): **db2 UNCATALOG DB <dbalias>**

Perform the following steps before you upgrade your DB2 server:

- Review upgrade recommendations and disk space requirements.
- Before upgrade, ensure that you have a valid backup image and valid log files for all databases taken in the previous release.
- After upgrade, ensure that you have valid log files for all databases taken in DB2 11.1.
- Back up all database manager configuration parameter values for each instance and all database configuration parameter values for each database.
- Perform other pre-upgrade tasks that apply to your environment.

Recovering through DB2 single partition server upgrade on Windows (detailed steps 3 of 3)

- **Administrator:** Upgrade instance to DB2 11.1: **db2iupgrade**
<instancename>
- **instance owner:** start instance: **db2start**
- **instance owner:** Re-catalog database(s) with recorded database path and authentication: **db2 CATALOG DB <dbname> AS <dbalias> ON <database-path> [AUTHENTICATION authentication-type]**
- **instance owner:** For database(s) that need recovery, continue roll forward until complete.
db2 ROLLFORWARD DB <dbalias> TO <isotime> [AND STOP]
- Once database comes out of roll forward pending state, examine all post upgrade tasks.

Use this procedure to recover one or more databases through a DB2 single partition Linux or UNIX server upgrade.

Preparation Steps: Reversing DB2 server upgrade

- Ensure have SYSADM, as well as root on Linux/UNIX or Local Administrator authority on Windows.
- Perform following steps before upgrading your DB2 server:
 - Review disk space requirements.
 - Take full backup of all databases that upgrading.
 - Back up all dbm cfg values for each instance and all db cfg values for each database.
- Keep existing pre-DB2 11.1 copy during upgrade of your DB2 server. To do this:
 - Select Install New option to create new copy when installing DB2 11.1.
 - Do not select Work with existing option
 - Then choose a pre-DB2 11.1 copy with upgrade action that is available on Windows.
- **Save all log files in active log paths to another directory in case you want to rollforward through these log files after reversing upgrade.**
- **Use db2cklog utility to determine what log files in active log paths are from pre-DB2 11.1 release. Record information for later use if rolling forward after reversing upgrade.**
- In multi-partitioned databases, must have offline backup for all database partitions.



115

© 2016 IBM Corporation

Reversing DB2 server upgrade involves creating a plan using the steps in this procedure to fall back to the DB2 release from which you upgraded your DB2 server. There is no utility to fall back to a previous release of DB2 database after upgrading your DB2 server.

Performing an upgrade in a test environment will help you identify any issues with the process and avoid having to reverse the upgrade.

Reversing DB2 server upgrade (1 of 2)

1. Log on to DB2 server as user with SYSADM authority.
2. Save all log files in active log paths (primary and mirror) in case need to rollforward through log files after reversing upgrade.
3. Drop all databases in DB2 11.1 by running DROP DATABASE.
4. Log on to DB2 server as root on Linux/UNIX or user with Local Administrator authority on Windows.
5. Drop DB2 11.1 instances by running db2idrop. Does not remove database files. Need to drop databases before dropping instances.
6. If upgraded pre-DB2 11.1 instances to DB2 11.1, re-create instances in pre-DB2 11.1 by running db2icrt. Then restore DBM CFG values for each instance using UPDATE DBM CFG.
7. For each pre-DB2 11.1 instance, log on to DB2 server as instance owner and restore upgraded databases from pre-DB2 11.1 full backup by running RESTORE DATABASE. Cannot upgrade databases from DB2 11.1 to pre-DB2 11.1 release.
8. If recreated instances using same instance owner had before upgrade and did not upgrade database to DB2 11.1 instance, database is still pre-DB2 11.1 release and can access it by just re-cataloging it.

REVERSE

To reverse a DB2 server upgrade, you need to perform the above steps.

In previous versions for Step 9, if restored pre-DB2 10.5 online backup and needed to rollforward through pre-DB2 10.5 log files had before upgrade, you had to rename all S*.MIG files in active log path using .LOG extension and issue ROLLFORWARD DATABASE.

Reversing DB2 server upgrade (2 of 2)

- If you have recoverable databases and want to rollforward through log files you had before upgrade, issue `ROLLFORWARD DATABASE`. Ensure you supply log files saved in step 2 to `ROLLFORWARD DATABASE` command. This can be done by copying log files back in to active log paths or supplying `OVERFLOW LOG PATH` on `ROLLFORWARD DATABASE` command. 
- **If using DB2 10.5 Fix Pack 7 or later, `ROLLFORWARD DATABASE` may return success or `SQL2463N`. Reissue command with `STOP` option to bring up database at end of pre-11.1 release.**
- **If using DB2 9.7, 10.1 or 10.5 Fix Pack 6 or earlier, `ROLLFORWARD DATABASE` may return success or `SQL1263N`. Rename error `S*.LOG` log file to `.NEW` extension and reissue command.**
- If multi-partition database, may need to repeat same steps for each partition or member. If using log archiving additional steps, such as manually retrieving log files to disk, may be required to ensure that rollforward utility does not replay log file from DB2 11.1.

To reverse a DB2 server upgrade, you need to perform the above steps.

Upgrading to a new DB2 11.1 Instance/Server (1 of 2)

- Perform a full **offline** database backup of your existing DB2 10.5, 10.1 or 9.7 database(s).
 - If database backups performed recently, can perform incremental *offline* database backup.
- Log on to new DB2 instance/server as root on Linux and UNIX or user with Local Administrator authority on Windows.
- Install DB2 11.1 on new DB2 instance/server.
- Create DB2 11.1 instances on new DB2 instance/server by running **db2icrt**.
- If similar resources, restore DBM CFG values for each instance.
- Transfer pre-DB2 11.1 backup files for all databases want to upgrade to new instance/server.
- Upgrade database to DB2 11.1 by using RESTORE DATABASE:
db2 RESTORE DATABASE <dbalias> FROM /db2/backups
- If performed incremental *offline* database backup before upgrade, must have access to most recent full *offline* database backup and incremental *offline* database backup and use automatic incremental restore to upgrade database.
 - Manual incremental restore will fail because each RESTORE DATABASE tries to upgrade database before database is completely recovered. 

**db2 RESTORE DATABASE <dbalias> INCREMENTAL AUTOMATIC
TAKEN AT *timestamp* WITHOUT PROMPTING**

- In multi-partitioned database, must execute RESTORE DATABASE on all database partitions starting with catalog database partition first.

If you want to upgrade to a new DB2 10.5 server, re-create your instances and then upgrade your databases by restoring a pre-DB2 10.5 database backup. After restoring the database backup, the RESTORE DATABASE command automatically runs the UPGRADE DATABASE command. A database backup from one platform family can only be restored on any system within the same platform family. For Windows operating systems, you can restore a database that was created on DB2 10.1 on a DB2 10.5 database system. For Linux and UNIX operating systems, as long as the endianness (big endian or little endian) of the backup and restore platforms is the same, you can restore backups that were produced on down level versions. The target system must have the same (or later) version of the DB2 database product as the source system. You cannot restore a backup that was created on one version of the database product to a system that is running an earlier version of the database product. For example, you can restore a DB2 10.1 on a DB2 10.5 database system, but you cannot restore a DB2 10.5 backup on a DB2 10.1 database system. Database configuration parameters are set to their defaults if the values in the backup are outside of the allowable range for the environment in which the database is being restored.

Upgrading to a new DB2 11.1 Instance/Server (2 of 2)

- If log file size is still not large enough, following error is returned:
 - SQL1704N Database upgrade failed. Reason code "3".
You must increase log file size and attempt to upgrade database again.
 - After database upgrade completed, reset size of log files to pre-upgrade values.
- Optional: Configure new instance/server to use new resources available to calculate buffer pool sizes, and DBM and DB CFG values. Run command to display only recommended values:


```
db2 CONNECT TO <dbalias> AUTOCONFIGURE
  USING MEM_PERCENT 80 WORKLOAD_TYPE complex
  NUM_STMTS 1 TPM 73 ADMIN_PRIORITY performance IS_POPULATED YES
  NUM_REMOTE_APPS 15 ISOLATION CS APPLY NONE;
```
- If choose not to run and apply, manually configure DB2 server to use new resources.
- Restore any external routines that backed up in pre-upgrade tasks.
- Verify database upgrade successful. Connect to upgraded databases and issue query:


```
cd samplefile-dir-clp
db2 connect to <dbalias>
```

```
db2 -tvf testdata.db2
  where samplefile-dir-clp is DB2DIR/samples/clp on Linux and UNIX and DB2DIR\samples\clp on
Windows, DB2DIR represents the location specified
during DB210.1 installation, and sample is database name.
```
- After upgrading DB2 instance/server, perform recommended post-upgrade tasks such as resetting diagnostic error level, adjusting log space size, and rebinding packages. In addition, verify that upgrade of DB2 instance/server was successful.

If you want to upgrade to a new DB2 10.5 server, re-create your instances and then upgrade your databases by restoring a pre-DB2 10.5 database backup. After restoring the database backup, the RESTORE DATABASE command automatically runs the UPGRADE DATABASE command.

You can restore a database from a backup image that was taken on a 32-bit level into a 64-bit level, but not vice versa. The DB2 backup and restore utilities should be used to back up and restore your databases. Moving a file set from one machine to another is not recommended as this can compromise the integrity of the database. In situations where certain backup and restore combinations are not allowed, you can move tables between DB2 databases using other methods:

- The db2move command
- The export command followed by the import or the load command

Upgrading new server using online backups from previous release

- Can rebuild database on previous release by using online database backups from the same release and then upgrade to DB2 11.1.
- Perform task only under following conditions:
 - If cannot upgrade existing instances and databases.
 - If did not perform full *offline* database backups recently or incremental *offline* database backups as indicated in Pre-upgrade tasks for DB2 servers.
- Transfer pre-DB2 11.1 online database backup files for all databases that want to upgrade to DB2 server.
- If have DB2 copy with same version as online database backups, install DB2 copy of same version. If performed online database backups from DB2 10.5 copy, must have DB2 10.5 copy installed on DB2 server.
- If do not have instance running on DB2 copy with same version as online backups, create instance under DB2 copy.
- As SYSADM, rebuild databases by:


```
RESTORE DB db-name REBUILD WITH ALL TABLESPACES IN DATABASE  
TAKEN AT timestamp-backup;  
ROLLFORWARD DB db-name TO END OF LOGS AND STOP;
```
- Can choose to rebuild database with just subset of table spaces, but must drop all table spaces in restore pending state after issue ROLLFORWARD DATABASE.
- Verify databases that rebuild are in consistent state by issuing GET DB CFG:


```
GET DB CFG FOR sample | FIND "consistent"
```

You can rebuild your database on a previous release by using online database backups from the same release and then upgrade to DB2 11.1.

Before upgrading your DB2 server, ensure that you have root access on Linux and UNIX operating systems or Local Administrator authority on Windows. All necessary full or incremental online pre-DB2 11.1 database backups of your databases so that you can rebuild your databases by using these online backups.

Perform this task only under the following conditions:

- If you cannot upgrade the existing instances and databases.
- If you did not perform full *offline* database backups recently or incremental *offline* database backups as indicated in Pre-upgrade tasks for DB2 servers.

Security Enhancements

▪ Native Encryption Supports Centralized Key Managers

- DB2 11.1 adds support for centralized key managers to store native encryption master keys.
- Can use any key manager product that implements Key Management Interoperability Protocol (KMIP) version 1.1 or greater
- Single centralized key manager can manage encryption keys for many databases.
- Validated on IBM's Security Key Lifecycle Manager (ISKLM).

DB2NIGHT SHOW
Episode #177
DB2 LUW V11.1 - Part 1
- OVERVIEW -
Berni Schiefer &
Matt Huras

▪ Row and column access control (RCAC) extended to column-organized tables

- DB2 11.1 extends RCAC to now cover column-organized tables .
- Row and column access control allows you to regulate data access at row level, column level, or both.
- RCAC complements table privileges model and ensures that your users have access to only the data that is required for their work.



A local or external key manager is typically used to manage the keys. A *data encryption key* is the encryption key with which actual user data is encrypted. A *master key* is a "key encrypting key": It is used to protect the data encryption key. Although the data encryption key is stored and managed by the database, the master key is stored and managed outside of the database.

The DB2 Version 10.5 Fix Pack 5 adds native database encryption to the DB2 database server. This enhancement is easy to implement and provides secure local key management that is based on Public Key Cryptography Standard #12 (PKCS#12). DB2 native encryption enables you to meet compliance requirements in a cost effective manner.

For details on how to implementing DB2 native database encryption see:

<http://www->

[01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/t0061766.html?lang=en](http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/t0061766.html?lang=en)

IBM BLU Enhancements Column Organized Tables (1 of 2)

- **Multiple Database Partition support**
- **Sorting using fast parallel radix sort on compressed & encoded data**
- **Improved SORTHEAP utilization**
- **NOT LOGGED INITIALLY can be used on CREATE/ALTER TABLE**
- **SQL OLAP improvements for deeper in-database analytics**
- **Additional Oracle Compatibility Support**
 - Wide rows and Logical character support (CODEUNITS32)
- **DGTT support (except not logged on rollback preserver rows)**
 - Parallel insert into not-logged DGTT from BLU source
- **IDENTITY and Expression generated columns**
- **European Language support (Code page 819)**
- **ROWID support**
- **Faster SQL Merge Support**
- **Nested Loop Join Support**
- **Row and Column Access Control (RCAC)**

DB2NIGHT SHOW
Episode #178
DB2 LUW V11.1 - Part 2
- BLU and Analytics -
John Hornibrook &
David Kalmuk

Starting with DB2 Version 11, you can create column organized tables in a Partitioned database environment. This allows you to leverage the enhanced support for analytic workloads introduced in DB2 10.5 at massive scale in clustered multi-server environments. Advantages include significantly reduced time-to-value and increased consumability, which you can achieve through minimal design requirements and reduced query tuning and debugging. Industry-leading compression, large performance gains for analytic queries, and large reductions in performance variation round out the benefits of deploying this technology.

The more processing that can be done without leaving the BLU engine, the better will the performance be.

DB2 11.1 implements a new high performance parallel sorting algorithm (fast radix sort) that works on column organized data. Not only does it occur within the BLU engine, it also works on the compressed data without decompression; and it is fast. As of now, this new sorting algorithm is only used with BLU, not for row based tables.

IBM BLU Enhancements Column Organized Tables (2 of 2)

- **Automatic Dictionary Creation during Insert/Ingest/Import to increase throughput for high concurrency workloads.**
 - Higher PCTENCODED (percent encoded) values in SYSCAT.COLUMNS
 - Better query performance
 - Increased throughput for high concurrency workloads

- **Improved performance and scalability for high concurrency workloads**
 - Highly concurrent workloads that have frequently-accessed common pages, such as small table lookups or frequent index access (especially via nested-loop joins) may see improved performance (lower transaction latency, lower latch and lock waits) and scalability (higher degree of concurrency) as result of these changes.

Prior to V 11.1, the only join method supported on columnar data was hash join (which is often the best choice for large amounts of data). However, even with analytical queries, filtering often reduces the qualifying data down to a small number of rows in one or both tables. In this case, a nest loop join (NL Join) may be better performing. NL Joins also support inequality join predicates, whereas hash joins do not.

There are a number of built-in functions that prior to V11.1 were not yet implemented within the BLU Engine. These include OLAP functions such as ROW_NUMBER, RANK, RATION_TO_REPORT, OLAP columnar functions such as AVG, MAX, MIN, and scalar functions like: LPAD, TO_CHAR, TO_NUMBER, MOD, MONTHNAME.

Internal improvements were made to the processing of the SQL Merge statement on columnar tables.

Being able to declare a column organized temp table also allows more processing within the BLU engine.

IBM AnalyticsIBM

pureScale Enhancements

- **Support for HADR Sync and NearSync modes**
- **Unified Workload Balancing**
 - User can explicitly define alternate members in a subset
 - Allows users currently using client affinity to move configuration to using member subsets so can exploit new benefits such as dynamic server side reconfiguration
- **Improved high availability and extended platform support in Geographically dispersed DB2 pureScale cluster (GDPC)**
- **GPFS Replication Up-and-Running and Operational Ease of Use**
- **Improved health checks for pureScale cluster: db2cluster -verify**
- **RHEL 7.2 support for DB2 pureScale**

DB2NIGHT SHOW
Episode #179
DB2 LUW V11.1 - Part 3
- OLTP and pureScale -
Steve Rees, Keri Romanufa, & Xun Xue

124© 2016 IBM Corporation

Support for SYNC and NEARSYNC for `hadr_syncmode` database configuration parameter has been added to pureScale.

Version 11.1 extends the configuration options for member subsets allowing the user to explicitly define alternate members in a subset. This allows users currently using client affinity to move their configuration to using member subsets so that they can exploit the new benefits such as dynamic server side reconfiguration.

GDPC extends its support on 10GE RoCE and TCP/IP network to cover all pureScale supported operating systems: AIX, SuSE Linux Enterprise Server (SLES) and Red Hat Enterprise Linux (RHEL). Each of these configuration now supports multiple adapter ports per member and CF to support higher bandwidth and improved redundancy at the adapter level. Furthermore, dual switches can be configured at each site to eliminate the switch as a site-specific single point of failure.

The configuration of GPFS replication in traditional DB2 pureScale cluster and Geographically Dispersed DB2 pureScale Cluster (GDPC) has been enhanced by integrating all management and monitoring task with the `db2cluster` utility

`db2cluster -verify` is a post-installation unified health check tool for a DB2 pureScale cluster. The validations performed include, but are not limited to, the following:

- Configuration settings in peer domain and GPFS cluster
- Communications between members and CFs
- Replication setting for each file system
- Status of each disk in the file system

DB2 pureScale is supported on RHEL 7.2 in Version 11.1.

Upgrade to 11.1 pureScale server (1 of 6)

- Log on to DB2 server as instance owner.
- Disconnect all applications and users on every member node.
db2 force applications all
db2 list applications
SQL1611W No data was returned by the Database System Monitor.
SQLSTATE=00000
- Stop all CLP sessions: **db2 terminate**
- Stop the database manager instance: **db2stop**
Note: Time needed for db2stop to complete depends on number of cluster caching facilities (CFs) and members in cluster (up to 10 minutes).
- If db2stop is not successful and fails to stop instance with timeout due to new incoming connections, run: **db2stop force**
- Stop all instance processes in members and cluster caching facilities (CFs) in any order by issuing on any node (hostname is name of each member or CF in cluster):
db2stop instance on <hostname>

Step-by-step approach for upgrading to DB2 11.1 pureScale Server.

Upgrade to 11.1 pureScale server (2 of 6)

- Install DB2 11.1 by performing following steps
 - A. Log on to DB2 server with root user authority to install DB2 11.1 code.
 - B. Put cluster management software into maintenance mode on all members and cluster caching facilities (CFs) by issuing command from pre-DB2 11.1 installation path . This command stops peer domain services on all hosts and prevents it from restarting during system maintenance. You can issue command only one time on any one of members or CFs in the cluster:
OLD_DB2_INSTALL_DIR/bin/db2cluster -cm -enter -maintenance -all
 - C. Put cluster file system into maintenance mode on all members and CFs by issuing command from pre-DB2 11.1 installation path. Command stops all hosts from accessing cluster files system (GPFS) during system maintenance. Issue command only one time on any one of members or CFs in the cluster.
OLD_DB2_INSTALL_DIR/bin/db2cluster -cfs -enter -maintenance -all
 - D. Install DB2 11.1 using DB2Setup wizard. Run command and select **Install New** option on Install a Product panel to install new copy of DB2 11.1.
db2setup -l /tmp/db2setup.log -t /tmp/db2setup.trc
In set up DB2 instance panel, select option **Do not create DB2 instance**. DB2 installer still performs installation, but can create instance later with db2icrt or db2isetup. DB2Setup wizard provides clear flow through which can start DB2 pureScale Feature installation from one member and successfully setup DB2 pureScale environment across multiple members. Cluster management software and cluster file system software are also upgraded during installation to meet DB2 11.1 requirements.

Step-by-step approach for upgrading to DB2 11.1 pureScale Server.

Upgrade to 11.1 pureScale server (3 of 6)

E. Ensure that GPFS cluster is at the minimum release level by using following steps:

- Use the `mmlsconfig` command to determine minimum release level and whether `usePersistentReserve` attribute is set to YES:
 - `root@XXX:~# /usr/lpp/mmfs/bin/mmlsconfig minReleaseLevel,usePersistentReserve minReleaseLevel 3.4.0.7 usePersistentReserve yes`
 - If `minReleaseLevel` value is 3.5 or higher or `usePersistentReserve` is set to NO, then go to Step F.
 - Use `mmchconfig` command to update GPFS configuration information to most current format that is supported by your GPFS level:
`/usr/lpp/mmfs/bin/mmchconfig release=LATEST`
Verifying that all nodes in the cluster are up-to-date
Verifying GPFS is stopped on all nodes
`mmchconfig: Command successfully completed`
`mmchconfig: Propagating the cluster configuration data to all affected nodes. This is asynchronous process.`
 - Verify that release level is updated, by issuing `mmlsconfig` command again:
`/usr/lpp/mmfs/bin/mmlsconfig minReleaseLevel minReleaseLevel 3.5.0.7.`

Step-by-step approach for upgrading to DB2 11.1 pureScale Server.

Upgrade to 11.1 pureScale server (4 of 6)

- F. As root, take cluster management software out of maintenance mode by issuing command from *pre-DB2 11.1 installation path*. Issue following command only one time on any one of the members or CFs in the cluster.

OLD_DB2_INSTALL_DIR/bin/db2cluster -cm -exit -maintenance -all

Note: If old DB2 installation path is DB2 10.1, must take cluster management software out of maintenance mode by issuing: OLD_DB2_INSTALL_DIR/bin/db2cluster -cm -exit -maintenance.

- G. Ensure that all of db2mnt resource groups are online, by issuing:

Issam -s "Name like %db2mnt%rg"

Online IBM.ResourceGroup:db2mnt-db2sd-rg Nominal=Online

├- Online IBM.Application:db2mnt-db2sd-rs

└- Online IBM.Application:db2mnt-db2sd-rs:XXX

└- Online IBM.Application:db2mnt-db2sd-rs:YYY

If any groups are reported as Unknown, then reenter and exit maintenance mode, as described in step above.

Issam -s "Name like %db2mnt%rg"

Unknown IBM.ResourceGroup:db2mnt-db2sd-rg Control=MemberInProblemState Nominal=Online

├- Online IBM.Application:db2mnt-db2sd-rs Control=MemberInProblemState

└- Online IBM.Application:db2mnt-db2sd-rs:XXX

└- Online IBM.Application:db2mnt-db2sd-rs:YYY

Note: In this scenario, need to repeat steps b and f only; do not need to repeat steps c, d, and e.

Step-by-step approach for upgrading to DB2 11.1 pureScale Server.

Upgrade to 11.1 pureScale server (5 of 6)

- F. As root, take cluster management software out of maintenance mode by issuing command from *pre-DB2 11.1 installation path*. Issue following command only one time on any one of the members or CFs in the cluster.

OLD_DB2_INSTALL_DIR/bin/db2cluster -cm -exit -maintenance -all

Note: If old DB2 installation path is DB2 10.1, must take cluster management software out of maintenance mode by issuing: OLD_DB2_INSTALL_DIR/bin/db2cluster -cm -exit -maintenance.

- G. Ensure that all of db2mnt resource groups are online, by issuing:

Issam -s "Name like %db2mnt%rg"

Online IBM.ResourceGroup:db2mnt-db2sd-rg Nominal=Online

├ Online IBM.Application:db2mnt-db2sd-rs

|└ Online IBM.Application:db2mnt-db2sd-rs:XXX

└ Online IBM.Application:db2mnt-db2sd-rs:YYY

If any groups are reported as Unknown, then reenter and exit maintenance mode, as described in step above.

Issam -s "Name like %db2mnt%rg"

Unknown IBM.ResourceGroup:db2mnt-db2sd-rg Control=MemberInProblemState Nominal=Online

├ Online IBM.Application:db2mnt-db2sd-rs Control=MemberInProblemState

|└ Online IBM.Application:db2mnt-db2sd-rs:XXX

└ Online IBM.Application:db2mnt-db2sd-rs:YYY

Note: In this scenario, need to repeat steps b and f only; do not need to repeat steps c, d, and e.

Step-by-step approach for upgrading to DB2 11.1 pureScale Server.

Upgrade to 11.1 pureScale server (6 of 6)

- As root, take cluster file system software out of maintenance mode by issuing command from pre-DB2 11.1 installation path.
OLD_DB2_INSTALL_DIR/bin/db2cluster -cfs -exit -maintenance -all
- As root, commit changes to cluster file system by issuing command from DB2 11.1 installation path. Issue following command only one time on any one of members or CFs in the cluster.
NEW_DB2_INSTALL_DIR/bin/db2cluster -cfs -commit
- As root, commit changes to cluster management software by issuing command from DB2 11.1 installation path. Issue following command only one time on any one of members or CFs in the cluster.
NEW_DB2_INSTALL_DIR/bin/db2cluster -cm -commit
- As instance owner, restart DB2 instance processes on all members and CFs with updated resources for cluster management software and cluster file system software by issuing command only one time on any one of the members or CFs in the cluster:
db2start instance on <hostname>
- Install all DB2 add-on products that were installed in DB2 copy from which upgrading.
- Upgrade DB2 pureScale instances. Refer to Knowledge Center "Upgrading DB2 pureScale instances".
- Upgrade databases. Refer to Knowledge Center "Upgrading databases".

Step-by-step approach for upgrading to DB2 11.1 pureScale Server.

Database Manager CFG Parameter Changes

Parameter Name	Details
MON_HEAP_SZ	Range of MON_HEAP_SZ extended on 64-bit instances from 0 - 2,147,483,647. MON_HEAP_SZ also changed from Uint16 type to Uint64 type.
INSTANCE_MEMORY	Can now specify INSTANCE_MEMORY limit by calculating percentage of available RAM divided by number of local partitions, or specify memory limit as number of 4KB pages..

- **AUTOMATIC:** Default. Computed value ranges between 75 percent and 95 percent of system memory.
- **1 - 100:** Specifies instance_memory limit by calculating percentage of available RAM divided by number of local partitions. In-memory value is updated at member startup time to reflect calculated number of 4KB pages. Used to set DB2's percentage consumption of total RAM on machine in DB2 instances with heterogeneous machine hardware configurations.
- **101 - system memory capacity:** Specifies memory limit as number of 4KB pages. Also represents tuning target if STMM is enabled.

131

© 2016 IBM Corporation

Take advantage of enhanced functionality or new features by adopting new functionality through the use of new database manager configuration parameters or new values for existing database manager configuration parameters.

AUTOMATIC: This is the default value. The AUTOMATIC setting results in a value that is computed at database partition activation. The computed value ranges between 75 percent and 95 percent of the system memory capacity on the system - the larger the system, the higher the percentage. For DB2 database products with memory usage restrictions, the computed value is also limited by the maximum that is allowed by the product license. For database partition servers with multiple logical database partitions, this computed value is divided by the number of logical database partitions.

For new database manager configuration parameters or changes to existing database manager configuration parameters that result in DB2 server behavior changes, adjust your existing applications or scripts.

Use the `db2pd -dbptnmem` command to show details on instance memory usage.

Use the new `ADMIN_GET_MEM_USAGE` table function to get the total instance memory consumption by a DB2 instance for a specific database partition, or for all database partitions. This table function also returns the current upper bound value

Database Configuration Parameter Changes

Parameter Name	Details
APPLHEAPSZ	Range of APPLHEAPSZ extended on 64-bit instances from 16 – 2,147,483,647. APPLHEAPSZ also changed from Uint16 type to Uint64 type.
CATALOGCACHE_SZ	Range of CATALOGCACHE_SZ extended on 64-bit instances from 8 - 2,147,483,647.
DBHEAP	Range of DBHEAP extended on both 32-bit and 64-bit instances from 32 – 2,147,483,647.
REC_HIS_RETENTN	Can now be configured online.
STAT_HEAP_SZ	Range of STAT_HEAP_SZ extended on 64-bit instances from 1,096 - 2,147,483,647.
STMTHEAP	Range of STMTHEAP extended on 64 bit instances from 128 - 2,147,483,647.

Take advantage of enhanced functionality or new features by adopting new functionality through the use of new database configuration parameters or new values for existing database configuration parameters.

For new database configuration parameters or changes to existing database configuration parameters that result in DB2 server behavior changes, adjust your existing applications or scripts.

Registry Variable Changes

- **DB2_DATABASE_CF_MEMORY**
 - CF self-tuning memory is set at the instance level, and is enabled by setting this registry variable to AUTO. This variable is enabled by default.
- **DB2_CPU_BINDING**
 - Can now use this variable to specify percentage of processors DB2 will bind to and its CPU processing consumption with MACHINE_SHARE option.
- **DB2_PMAP_COMPATIBILITY**
 - Default value has changed from ON to OFF.

Take advantage of enhanced functionality or new features by using new registry variables or new values for registry variables.

DB2_PMAP_COMPATIBILITY; Operating system: All; Default: ON, Values: ON or OFF. This variable allows users to continue using the `sqlgtpi` and `sqlugrpn` APIs to return, respectively, the distribution information for a table and the database partition number and database partition server number for a row. The default setting, ON, indicates that the distribution map size remains 4 096 entries (the pre-Version 9.7 behavior). When this variable is set to OFF, the distribution map size for new or upgraded databases is increased to 32 768 entries (the Version 9.7 behavior). If you use the 32K distribution map, you need to use the new `db2GetDistMap` and `db2GetRowPartNum` APIs. Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the `db2set` command with the `-immediate` parameter.

For new registry variables or changes to registry variables that result in DB2 server behavior changes, adjust your applications or scripts.

New Registry and Environment Variables

▪ **DB2_USE_FAST_LOG_PREALLOCATION**

- AIX, Linux on Veritas VxFS, JFS2, GPFS, ext4 (Linux only) and xFS (Linux only) file systems
- Default: OFF, **ON under DB2_WORKLOAD=SAP**, Values: ON or OFF
- Allows the fast preallocation file system feature to reserve space for log files, and speed up the process of creating or altering large log files, if the underlying file system supports this feature. This speed improvement is implemented at a small delta cost of performing actual space allocation during runtime when log records get written to such preallocated log files.

▪ **DB2_BCKP_COMPRESSION**

- Specifies whether backups are compressed with common DB2 compression or with NX842 compression. NX842 compression is only available on AIX.
- Default: OFF, COMPRESS or NX842

▪ **DB2_SYMPHONY_WLM**

- Introduced in DB2 10.5.0.4, informs DB2 instance that it is running in an integrated cluster which involves mixed technologies such as Platform Symphony and DB2.

134

© 2016 IBM Corporation

DB2_USE_FAST_LOG_PREALLOCATION: Operating system: AIX and Linux on Veritas VxFS, JFS2, GPFS, ext4 (Linux only) and xFS (Linux only) file systems.

Default: OFF, ON under DB2_WORKLOAD=SAP, Values: ON or OFF.

Allows fast preallocation file system feature to reserve space for log files, and speed up process of creating or altering large log files. This speed improvement is implemented at a small delta cost of performing actual space allocation during runtime when log records get written to such preallocated log files.

Hardware accelerated backup and log file compression

- By using nest accelerator NX842 of POWER 7+ and POWER 8 processors, can achieve hardware compression for backup images and log archive files on AIX V7 TL3 SP3 and AIX V6 TL9 SP3.
 - very fast compression through special hardware compression unit NX842 on POWER CPUs. General CPU resources are not used for this compression. NX842 compression units typically not used for AME on database servers since deep row compression, adaptive compression and index compression can make memory compression inefficient. Compression algorithm in hardware provides faster compression than the common DB2 compression.
- To start a backup, it is necessary to specify library:


```
db2 backup db <alias> compress comprlib libdb2nx842.a
```
- Backups can be compressed by default with NX842.


```
db2set DB2_BCKP_COMPRESSION=NX842
db2 backup db <alias> compress
```
- To activate NX842 hardware compression for log archive compression:


```
db2 update db cfg for <alias> using LOGARCHCOMPR1 NX842
```



Active Memory Expansion (AME) has to be licensed but must not be enabled. This is a temporary restriction and not technical limitation. Active Memory Sharing (AMS) has to be deactivated on logical partition (LPAR). CPU has to be POWER 7+ or later.

The NX842 hardware compression can also be used for log archive compression. To activate this, change the database configuration parameter LOGARCHCOMPR1 or LOGARCHCOMPR2 to NX842 using this command.

The registry variable db2set DB2_BCKP_COMPRESSION=COMPRESS or NX842 specifies whether backups are compressed with common DB2 compression or with NX842 compression. NX842 compression is only available on AIX.

CREATE TABLE Extensions

- **CREATE TABLE can now use SELECT clause to generate table definition and populate table with data**
 - CREATE TABLE T1 AS (SELECT * FROM TAB1) WITH DATA
- **Previously: CREATE TABLE can also override column names and create table definition only**
 - CREATE TABLE T1 (ZIP) AS (SELECT ZIP_CODE FROM TAB1) DEFINITION ONLY
 - CREATE TABLE T1 LIKE TAB1
 - Does not allow you to override column names or select specific columns
- **Can populate data directly into new table but beware of logging!**
 - CREATE TABLE T1 (ZIP, STATE) AS
(
 SELECT ZIP_CODE, STATE_CODE FROM TAB1
 WHERE DIVISION = 19
) WITH DATA



New CREATE TABLE extensions allow you to generate the table definition and populate table with data. The WITH DATA option is logging and is one unit of work so beware of running out of active log space.

ADMIN_MOVE_TABLE new options

- **REPORT** option calculates set of values to monitor progress of single or multiple table moves.
 - Focus is COPY and REPLAY phase of running table move.
 - To get values for all table moves, tabschema and tablename must be NULL or empty string.
 - Monitoring values inserted into Declared Global Temporary Table named SESSION.ADMIN_MOVE_TABLE
 - Result set to monitor a single table move:
 - STATUS, AGENT_ID, INIT_START, COPY_THROUGHPUT, COPY_ECT, ROWS_STAGING, REPLAY_THROUGHPUT, INFLOW_STAGING, OUTFLOW_STAGING, GROWTH_STAGING, REPLAY_ECT
 - Result set to monitor all table moves initiated:
 - TABSCHEMA, TABNAME, STATUS, AGENT_ID, COPY_ECT, GROWTH_STAGING
- **TERM** kills running table move, roll backs all open transactions and sets table move to well defined operation status. Then table move can be canceled or continued.

The ADMIN_MOVE_TABLE now includes two new options: REPORT and TERM. The REPORT option can be used to monitor the progress of table moves. The TERM option can be used to terminate a table move in progress.

If operation REPORT is used, the result set is a subset of the DGTT and depends on reporting a single table move or all table moves initiated.

COPY_THROUGHPUT is Estimated COPY completion time. The values is calculated using COPY_THROUGHPUT, COPY_TOTAL_ROWS and CARDNULL, if calculation is not possible or the COPY phase has ended. NULL, if calculation is not possible or the COPY phase has ended.

ROWS_STAGING is Number of rows in the staging table. Calculated using MON_GET_TABLE values rows_inserted - rows_deleted. Unit: rows. NULL, if calculation is not possible.

REPLAY_THROUGHPUT - REPLAY performance. Formula: REPLAYED_TOTAL_ROWS/REPLAY_TOTAL_TIME. Unit: rows/second

NULL, if calculation is not possible.

REPLAY_ECT - Estimated REPLAY completion time. Special value 9999-12-31-23:59:59 indicates that under current progress and workload conditions, the move will not end. NULL, if calculation is not possible.

INGEST Enhancements

DELIMITER PRIORITY CHAR

- Current default priority for delimiters is record delimiter, character delimiter, column delimiter. This modifier protects existing applications that depend on older priority by reverting delimiter priorities to character delimiter, record delimiter, column delimiter.
 - Equivalent to `delprioritychar` modifier on LOAD and IMPORT.

NO CHAR DELIMITER

- When you use INGEST utility, all bytes that are found between the column delimiters are considered to be part of the column's data. Character delimiters are parsed as part of column data. Do not specify this option if the data was exported by using a DB2 database system (unless `nochardel` modifier was specified during export). The `nochardel` modifier is provided to support vendor data files that do not have character delimiters. Improper usage might result in data loss or corruption.
- Equivalent to the `nochardel` modifier on LOAD and IMPORT.

RADIX POINT IMPLIED

- Location of implied decimal point is determined by column definition; it is no longer assumed to be at end of value. For example, value 12345 is ingested into `DECIMAL(8,2)` column as 123.45, not 12345.00. The `PACKED` decimal option and the `RADIX POINT` option are not compatible (SQLCODE -3525).
- Equivalent to `implieddecimal` modifier on LOAD and IMPORT.

DEFAULTIF

- Existing option now accepts NULL as a parameter.

The INGEST Utility has several enhancements in DB2 11.1

Additional DB2 Command Enhancements

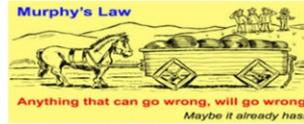
▪ Backup Database new NO TABLESPACE option

NO TABLESPACE backup does not contain table spaces; only history file.

db2 "BACKUP DB <alias> NO TABLESPACE to /histbkup"

– Image has:

- Table space granularity
- 'P' in History File
- No table spaces listed



db2 "RESTORE DB <alias> FROM /histbkup HISTORY FILE"

▪ Can access IBM SoftLayer Object Storage or Amazon Simple Storage Service (S3) directly with INGEST, LOAD, BACKUP, and RESTORE by using storage access aliases.

– **LOAD FROM <remotefilename> ...**

- The remotefilename refers to file that is on remote storage, such as IBM SoftLayer Object Storage or Amazon Simple Storage Service (S3), and is being accessed using storage access alias.
- Syntax of remote file names: **DB2REMOTE://<alias>//<storage-path>/<file-name>**

The NO TABLESPACE option on the backup database command allows you to backup the RHF and meta data for the database without backing up any table space extents.

This backup is only a minimal size and can be restored quickly. It also allows you to restore the history file within the storage system after a flash copy or other storage based backup process.

The NO TABLESPACE option gets the configuration information, history file details and other DB2 system information which can be restored to a HADR standby database after a takeover and it assumes the role of a primary database.

New db2relocatedb option generates script

- **-g generates script**

db2relocatedb -g relocate.cfg -d <alias>

```
DB_NAME=oldName,oldName
DB_PATH=oldPath,oldPath
INSTANCE=oldInst,oldInst
NODENUM=nodeNumber
LOG_DIR=oldDirPath,oldDirPath
CONT_PATH=oldContPath1,oldContPath1
...
STORAGE_PATH=oldStoragePath1,oldStoragePath1
...
```



- **-r Replace Definition replaces strings in generated script**

db2relocatedb -g relocate.cfg -d PRD -r PRD=QAS,db2prd=db2qas

```
DB_NAME=PRD,QAS
DB_PATH=/db2/PRD,/db2/QAS
INSTANCE=db2prd,db2qas
NODENUM=0
STORAGE_PATH=/db2/PRD/sapdata1,/db2/QAS/sapdata1
STORAGE_PATH=/db2/PRD/sapdata2,/db2/QAS/sapdata2
```

Generates a configuration file and specifies the name of the file containing the configuration information. This can be a relative or absolute file name. Without the option -r, the output looks as above in the first output.

If you want to relocate this database, to change the database name to QAS, to use the instance db2qas, and to change the autostorage paths accordingly, you can use the above command. The output looks like the second output.

INPLACE table REORG support for Range Partitioned Tables

- Single data partition of data partitioned table can now be reorganized with the INPLACE option if the table has no nonpartitioned indexes.
- Only one data partition can be reorganized at a time.
- INPLACE table reorganization can be run only on tables that are at least three pages in size.

```
REORG TABLE T1 INPLACE  
ON DATA PARTITION <partition name>
```

The manageability of large range partitioned tables has been improved. A single partition of a partitioned table can now be reorganized with the INPLACE option if the table has no nonpartitioned indexes.

INPLACE reorg requires no need for any other additional storage for the reorg and its partition.

The INPLACE reorg can be monitored through the usual reorganization monitoring functions and db2pd functions.

New Troubleshooting command parameters

Troubleshooting Command	New Parameter
db2pd	<ul style="list-style-type: none"> • detail - suboption to db2pd -locks detail. Displays TableNm, SchemaNm, AppNode for applications locks being waited on. • -extentmovement - displays extent movement status of your database. • -membersubsetstatus - dumps the state of member subsets. • -subsetid - identify a subset
db2support	<ul style="list-style-type: none"> • -alldbs alldatabases - specifies that information about all databases that are found in the database directory is collected. • -sg group_name -system_group group_name - collect system-related information about the specified system group. • -su user_name -system_user user_name - collect system-related information about the specified system user. • -wlm - collect info related to WLM issues as part of optimizer mode with collection level 0 and above.

There are new options on the db2pd and db2 support commands in DB2 11.1.

dsmtop - DB2 text-based monitoring tool command

- **Lightweight, low-overhead**, text-only environment such as putty window or simple UNIX command line. No need for web server or windowing environment.

```
dsmtop -d sample -n localhost -r 50000 -u mel
```

- Monitoring accomplished by using mon_get table functions.
- **Can monitor DB2 10.1 and above.**
- Intended primarily for use on Linux and AIX, but **will run on Windows** with some limitations. Uses JDBC type 4 driver.
- Look and feel similar to now deprecated db2top command. Provides:
 - Sessions: See at a glance which connections are active, blocked, or idle.
 - Running SQL: See a list of recently run statements. Drill down is provided to see the full SQL text or run explain on a statement.
 - Top Consumers: Find which connections or activities are consuming the most CPU, IO or other resource.
 - **Time spent: Shows a breakdown of where the monitored database is spending time.**

The tool dsmtop provides similar DB2 db2top metrics using the new lightweight in-memory facilities. Also, dsmtop provides increased time-spent metrics for all the DB2 options including DB2 BLU and pureScale, additional REORG information, menus for DB2 DBA beginners, and is now supported on the Windows platform.

The dsmtop tool can also be used against DB2 LUW V10.1 and V10.5 as well as DB2 LUW V11 features.

See the following article for an overview of dsmtop:

<https://www.ibm.com/developerworks/community/groups/service/html/communitystart?communityUid=e0c78988-2114-4609-8ad0-69b71e04e75f>

There is a wiki containing dsmtop information:

<https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Text%20Based%20Monitoring%20for%20DB2%20LUW%20-%20dsmtop/page/Welcome%20to%20Text%20Based%20Monitoring%20for%20DB2%20LUW%20-%20dsmtop>

Inline Optimization Guidelines (Hints)

- Optimization guidelines can be used to specify the access plan
- Can specify using an XML document stored in SYSTOOLS.OPT_PROFILE table
 - Enable through bind option, special register
- DB2 11.1 - optimization guidelines can be specified with SQL statement in a SQL comment
 - Only one set of guidelines can appear in a /* */ comment after entire SQL statement
 - No registry variable setting required

```
SELECT T.NAME FROM TABLE1 T WHERE T.NAME = 'MEL'
/* <OPTGUIDELINES><IXSCAN TABLE='T'
INDEX='T.IX2'></OPTGUIDELINES> */
```

Another enhancement in DB2 11.1 is the new convention and configuration for SQL Optimization Guidelines or Hints. SQL Optimization Guidelines are used to lock in access paths for performance sensitive SQL statements that need particular access paths. The Inline SQL Optimization Guidelines convention is simpler because it no longer requires the use of the registry variable.

SQL Optimization Guidelines can now be implemented by adding optimization guidelines after the SQL statement between “/* */”. Within the SQL Optimization Guidelines a standard XML convention is used to detail a single Hint for the SQL statement’s access path. Adding the SQL Optimization Guideline directly after the SQL statement provides more transparency.

Troubleshooting & problem determination enhancements

DBMS_UTILITY.FORMAT_CALL_STACK and DBMS_UTILITY.FORMAT_ERROR_BACKTRACE

- These utility functions can be used in application code, as part of error reporting strategy, or for problem determination in development environment.
- 
- Work in tandem as part of error or exception handling strategy during development. Developers can insert functions into application's error handling code at points where they anticipate errors might occur:
 - `FORMAT_CALL_STACK` returns a formatted view of stack at time a routine is called.
 - `FORMAT_ERROR_BACKTRACE` returns similar view at time of last error in compiled routine.

These features help determine the source of an error in a list of active routines, including SQL and external routines, that were running at the time an SQL error took place.

PL_STACK_TRACE database facility

- When SQL PL and PL/SQL error stack logging configuration parameter is set to **NONE**, error stack logging is disabled; no error stack trace records will be written to db2diag.log.
- When set to **ALL**, error stack logging enabled. Error stack trace records written to db2diag.log file each time a new SQL error detected within SQL PL or PL/SQL routine.
- When set to **UNHANDLED**, error stack logging enabled. Error stack trace records written to db2diag.log file each time a new SQL error detected within SQL PL or PL/SQL routine, **but only if** error is not captured by SQL PL condition handler or PL/SQL exception handler within SQL routine that is running currently.
- **db2 update db cfg for <dbalias> using PL_STACK_TRACE [NONE; ALL; UNHANDLED]**



Starting in DB2 Version 10.5 Fix Pack 7, this parameter determines whether error stack logging is enabled or disabled for SQL PL and PL/SQL routines.

This facility can be set at the database level to log SQL errors in procedural code. Logs are formatted to show the complete call stack by routine name and source line number for all active SQL and external routines. This facility is intended for production environments that are not equipped to handle errors, and where changes to application code are impractical.

Application Development Changes (1 of 3)

▪ User-defined objects called LOCALTIMESTAMP

- LOCALTIMESTAMP special register synonym added to increase portability across IBM database products
 - LOCALTIMESTAMP can be synonym for CURRENT TIMESTAMP.
 - LOCALTIMESTAMP(integer) can be synonym for CURRENT TIMESTAMP(integer).
- If user-defined object (column, user-defined variable, user-defined function) has name LOCALTIMESTAMP, an unqualified reference to this object resolves to special register instead of the user-defined object.
- **Resolution:** You must fully qualify all references to these objects or rename them. You can also use a delimited identifier for columns.

▪ HEXTORAW function

- Starting in DB2 11.1, return data type of HEXTORAW function is VARBINARY data type.
- Previously HEXTORAW function was a synonym for the VARCHAR_BIT_FORMAT scalar function and return data type was VARCHAR FOR BIT DATA. No longer true in 11.
- **Resolution:** If require that HEXTORAW returns VARCHAR FOR BIT DATA, explicitly cast the result.
 - Consider existing MQTs that call HEXTORAW in select list of the fullselect that defines the MQT. If get SQL0344N when access MQT, drop and re-create affected MQT.

In 11.1, the LOCALTIMESTAMP special register synonym is added to increase portability across IBM database products. If a user-defined object such as a column, a user-defined variable, or a user-defined function has the name LOCALTIMESTAMP, an unqualified reference to this object resolves to the special register instead of the user-defined object. You must fully qualify all references to these objects or rename them. You can also use a delimited identifier for columns.

The HEXTORAW function returns a bit string representation of a hexadecimal character string. The result of the function is a VARBINARY. The length attribute of the result is half the length attribute of character-expression and the actual length is half the length of the actual length of character-expression. If the first argument can be null, the result can be null; if the first argument is null, the result is the null value. To represent a string of hexadecimal characters in binary form:

HEXTORAW('ef01abC9')

The result is a VARBINARY(4) data type with a value of BX'EF01ABC9'.

Application Development Changes (2 of 3)

▪ Reduce cost of failback: affinityFailbackInterval changes

- When `affinityFailbackInterval` is set, current server first disconnects, then tries to connect to primary server. But if primary server is down, connection is re-established to current server once again.
- **Resolution:** With new changes, current server first pings primary server and only if primary server is up, then current server is disconnected and connection is established to primary server.

▪ Attribute to efficiently set global variables

- Currently, users can set global variables in two ways:
 1. Mentioning global variable names and its corresponding values inside `sessionglobalvariables` section of `db2dsdriver.cfg`.
 - Session global variables flow to server along with first SQL statement that is used after connection established.
 2. Use `SQLExecDirect` API by invoking `SET sql`.
 - Only way to set global variables post-connection. However, invoking `SET sql` using `SQLExecDirect()` API will flow these variables immediately to the database server.
- **Resolution:** In DB2 11.1, new mechanism provided to chain global variables along with next SQL statement and flow to server instead of flowing immediately thereby reducing network trips.

Additional IBM® data server driver configuration keywords for high availability support are also available. However, they are not listed under the IBM data server driver configuration keywords section because they cannot be used individually.

You are required to use a set of keywords depending on the high availability feature that you want to enable. The `affinityFailbackInterval` parameter is the number of seconds to wait after the first transaction boundary to fail back to the primary server. Set this value if you want to fail back to the primary server. Fail back will only be attempted if the primary server is reachable. The default value is 0, which means that no attempt is made to fail back to the primary server.

Application Development Changes (3 of 3)

▪ Support for CHAR(255)

- CHAR data type now supports new maximum length of 255 bytes (previously 254 bytes).
 - can be used wherever CHAR types are specified, including tables, views, host variables, and also user-defined types and functions.
 - Change is affecting return type of scalar function CONCAT when combined length is 255 bytes.
 - Change also affects result length of scalar function CHAR without an explicit length parameter.
- **Resolution:** Consider existing MQTs that call CONCAT or CHAR in the select list of the fullselect that defines the MQT. If get error SQL0344N when access MQT, drop and re-create the affected MQT.
 - Host variables that carry result of call to CHAR function without explicit length might have to be increased in length to avoid buffer overflow.

Changed functionality typically involves changes in default values or an outcome different from what would have occurred in previous releases. For example, an SQL statement that you used in a previous release might produce different results in 11.1.

Maintaining application compatibility across releases is a key priority. However, some behavior has to change to take advantage of new and changed functionality in the current release.

Application development enhancements

- **Microsoft Application Virtualization (App-V) Sequencer**

- The App-V client presents user with neat, locally-installed application experience for virtualized applications.
- DB2 11.1 provides sequencing, configuration and installation steps to create App-V package for IBM Data Server Driver using Microsoft App-V sequencer.



- See "Virtualizing the IBM Data Server Driver on Windows platforms" in Knowledge Center for details

The IBM Application Virtualization (MS App-V) platform allows applications to be deployed in real-time to any client from a virtual application server. It removes the need for traditional local installation of applications and presents the user with a neat, locally-installed application experience for virtualized applications.

SQL compatibility enhancements (1 of 7)

- **BINARY and VARBINARY data types support binary string data**
 - BINARY and VARBINARY data types allow binary string data to be stored and manipulated without overhead of using a BLOB type.
 - A binary string is a sequence of bytes that are used to store as pictures, sound, or mixed media.
 - BINARY and VARBINARY data types are compatible with each other and are compatible with the BLOB data type.
 - Binary string data types are not compatible with character string data types, except those character strings that are defined as FOR BIT DATA.
 - Support for BINARY and VARBINARY data types enhances compatibility with other relational database management systems.

A *binary string* is a sequence of bytes. Unlike character strings, which usually contain text data, binary strings are used to hold non-traditional data such as pictures, voice, or mixed media.

Binary strings are not associated with a code page. The length of a binary string is the number of bytes. Only character strings of the FOR BIT DATA subtype are compatible with binary strings.

The unit of length for the binary string data type is OCTETS. The unit of length for the binary string data type cannot be explicitly specified.

There are two types of varying-length binary string:

- A VARBINARY can be up to 32 672 bytes long.
- A BLOB (*binary large object*) can be up to 2 gigabytes minus 1 byte (2 147 483 647 bytes) long.

All values in a fixed-length string column have the same length, which is determined by the length attribute of the data type. The length attribute must be 1 - 254, inclusive.

SQL compatibility enhancements (2 of 7)**▪ Regular expression support in scalar functions enhance search ability**

- REGEXP_COUNT
- REGEXP_EXTRACT
- REGEXP_INSTR
- REGEXP_LIKE
- REGEXP_MATCH_COUNT
- REGEXP_REPLACE
- REGEXP_SUBSTR

▪ Extensions that enhance SQL compatibility

- Functions added to OLAP specification:
 - NTH_VALUE
 - CUME_DIST
 - PERCENT_RANK
- offset clause added to subselect
- DISTINCT clause added to LISTAGG aggregate function

▪ Support for outer joins using outer join operator, which is plus sign (+) enhances cross-vendor support

- Queries can use outer join operator (+) as alternative syntax within predicates of WHERE clause.
- Use only when enabling applications from database products other than DB2.
- Enable by setting DB2_COMPATIBILITY_VECTOR=04.

The CUME_DIST function is a distribution function that returns a cumulative distribution of a row within an OLAP window, expressed as a value between 0.0 - 1.0. The result is computed as follows: The number of rows preceding or peer with the current row in the OLAP window, divided by the number of rows in the OLAP window. The data type of the result is DECFLOAT(34). The result cannot be NULL.

The PERCENT_RANK function is a distribution function that returns a relative percentile rank of a row within an OLAP window, expressed as a value between 0.0 - 1.0. When the number of rows in the OLAP window is greater than 1, the result is computed as follows: The RANK of the current row in the OLAP window minus 1 divided by the number of rows in the OLAP window minus 1. Otherwise, the result is 0.0. The data type of the result is DECFLOAT(34). The result cannot be NULL.

The NTH_VALUE function returns the expression value for the nth-row row in an OLAP window. The NTH_VALUE function returns the expression value for the nth-row row in an OLAP window.

- expression - An expression that specifies the current row in an OLAP window. The expression must return a value that is a built-in data type. (SQLSTATE 42884).
- nth-row - An expression that specifies which row of the OLAP window to return. The expression must return a value that is a built-in numeric data type, a CHAR, or a VARCHAR data type. In a Unicode database, the expression can also be a GRAPHIC or VARGRAPHIC data type. CHAR, VARCHAR, GRAPHIC, and VARGRAPHIC are supported using implicit casting. If the expression is not a SMALLINT, INTEGER, or BIGINT, it is cast to BIGINT before the function is evaluated. The value must be greater than 0 (SQLSTATE 22016). The expression must be a constant, a variable, or a cast of a constant or variable (SQLSTATE 42819).
- FROM FIRST or FROM LAST - Specifies how nth-row is applied. If FROM FIRST is specified, nth-row is treated as counting forward from the first row in the OLAP window. If FROM LAST is specified, nth-row is treated as counting backward from the last row in the OLAP window.
- RESPECT NULLS or IGNORE NULLS - Specifies how NULL values in the OLAP window are handled. If RESPECT NULLS is specified, all rows where the expression value for the row is the null value are considered in the calculation. If IGNORE NULLS is specified, all rows where the expression value for the row is the null value are not considered in the calculation. The data type of the result of NTH_VALUE is the same as the data type of expression. The result can be null. If nth-row is null, the result is the null value. If the number of rows in the OLAP window (including null values if RESPECT NULLS is specified or excluding null values if IGNORE NULLS is specified) is less than the value of nth-row, the result is the null value. The NTH_VALUE function is a non-deterministic function because the window-order-clause is not required and when window-order-clause is

specified, rows with equivalent sort keys have a non-deterministic order.

The offset-clause specifies the number of rows to skip before any rows are retrieved. Use this clause to communicate to the database manager that the application does not start retrieving rows until offset-row-count rows are skipped. If offset-clause is not specified, the default is equivalent to OFFSET 0 ROWS. An attempt to skip more rows than the number of rows in the intermediate result table is handled the same way as an empty result table. The offset-clause must be followed immediately by a fetch-first-clause.

Determining a predictable set of rows to skip requires the specification of an ORDER BY clause with sort keys that would uniquely identify the sort order of each row in the intermediate result table. If the intermediate result table includes duplicate sort keys for some rows, the order of these rows is not deterministic. If there is no ORDER BY clause, the intermediate result table is not in a deterministic order. If the order of the intermediate result table is not deterministic, the set of skipped rows is unpredictable.

offset-row-count - An expression that specifies the number of rows to skip before any rows are retrieved. The expression must evaluate to a numeric value that is constant for the query and therefore it cannot contain column names (SQLSTATE 428H7). The numeric value must be a positive number or zero (SQLSTATE 2201X). If the data type of the expression is not BIGINT, the result of the expression is cast to a BIGINT value.

If the fullselect contains an SQL data change statement in the FROM clause, all the rows are modified regardless of the number of rows to skip.

The keywords ROW and ROWS can be used interchangeably. The result is unchanged; however, using ROWS is more readable when associated with a number of rows other than 1.

The LISTAGG function aggregates a set of string elements into one string by concatenating the strings. Optionally, a separator string can be provided which is inserted between contiguous input strings. The function is applied to the set of values that are derived from the first argument by the elimination of null values. If DISTINCT is specified, duplicate string-expression values are eliminated. If a separator argument is specified that is not the null value, the value is inserted between each pair of non-null string-expression values. If DISTINCT is specified for LISTAGG, the sort-key of the ORDER BY specification must match string-expression (SQLSTATE 42822). If string-expression is implicitly cast, the sort-key must explicitly include a corresponding matching cast specification. If DISTINCT is specified for LISTAGG, the sort-key of the ORDER BY specification must exactly match string-expression (SQLSTATE 42822).

Queries can use the outer join operator (+) as alternative syntax within predicates of the WHERE clause. You should use the outer join operator only when enabling applications from database products other than the DB2 product to run on a DB2 database system.

Enable the outer join operator support by setting the DB2_COMPATIBILITY_VECTOR registry variable to hexadecimal value 0x04 (bit position 3), and then stop and restart the instance to have the new setting take effect. To take full advantage of the DB2 compatibility features for Oracle applications, the recommended setting for the DB2_COMPATIBILITY_VECTOR is ORA, which sets all of the compatibility bits.

```
SELECT * FROM T1 LEFT OUTER JOIN T2 ON T1.PK1 = T2.FK1 AND T1.PK2 = T2.FK2
```

is equivalent to

```
SELECT * FROM T1, T2 WHERE T1.PK1 = T2.FK1(+) AND T1.PK2 = T2.FK2(+)
```

SQL compatibility enhancements (3 of 7)

- **New CREATE FUNCTION allows you to create own aggregate functions**

- Returns a single value that is result of evaluation of set of like values such as those in a column within a set of rows
- Use your choice of programming languages
- Four sections within the function are defined based on stage of the aggregation process
 - INITIATE, ACCUMULATE, MERGE, and FINALIZE

- **New Data Types**

- INT2, INT4, INT8, FLOAT4, FLOAT8, BPCHAR

- **OVERLAPS predicate determines whether 2 chronological periods overlap**

- (--start1--,--end1--)--OVERLAPS--(--start2--,--end2--)

- **Syntax alternatives**

SYNTAX	SYNTAX ALTERNATIVE
LIMIT ... OFFSET	FETCH FIRST ... OFFSET
ISNULL and NOTNULL	IS NULL and IS NOT NULL
DISTRIBUTE ON	DISTRIBUTE BY

The CREATE FUNCTION (aggregate interface) statement is used to register a user-defined aggregate function at the current server. An aggregate function returns a single value that is the result of an evaluation of a set of like values, such as those in a column within a set of rows.

The offset-clause specifies the number of rows to skip before any rows are retrieved. Use this clause to communicate to the database manager that the application does not start retrieving rows until offset-row-count rows are skipped. If offset-clause is not specified, the default is equivalent to OFFSET 0 ROWS. An attempt to skip more rows than the number of rows in the intermediate result table is handled the same way as an empty result table. The offset-clause must be followed immediately by a fetch-first-clause.

The OVERLAPS predicate determines whether two chronological periods overlap. A chronological period is specified by a pair of date-time expressions (the first expression specifies the start of a period; the second specifies its end).

SQL compatibility enhancements (4 of 7)

OFFSET with FETCH FIRST

- Can be specified before **FETCH FIRST** clause
- Rows are retrieved after **OFFSET** values are skipped
 - SELECT * FROM T1
 OFFSET 10 ROWS FETCH FIRST 5 ROWS ONLY
- **OFFSET** can be used in subselect
 - SELECT * FROM T1
 WHERE C1 >
 (SELECT C1 FROM TAB1
 OFFSET 10 FETCH FIRST 1 ROW ONLY)

OFFSET clause can be specified before the FETCH FIRST clause on a SELECT.

SQL compatibility enhancements (5 of 7)**▪ DATASLICEID pseudo column**

- Any unresolved and unqualified column reference to DATASLICEID pseudocolumn is converted to NODENUMBER function and returns database partition number for a row.
- Count number of instances in which row for given employee in EMPLOYEE table is on different database partition from description of employee's department in DEPARTMENT table:

```
SELECT COUNT(*) FROM DEPARTMENT D, EMPLOYEE E
WHERE D.DEPTNO=E.WORKDEPT
AND E.DATASLICEID <> D.DATASLICEID
```

- Joins EMPLOYEE and DEPARTMENT tables so that rows of two tables are on same database partition:

```
SELECT * FROM DEPARTMENT D, EMPLOYEE E
WHERE E.DATASLICEID = D.DATASLICEID
```

Any unresolved and unqualified column reference to the DATASLICEID pseudocolumn is converted to NODENUMBER function and returns the database partition number for a row. For example, if DATASLICEID is used in a SELECT clause, the database partition number for each row is returned in the result set.

The specific row (and table) for which the database partition number is returned by DATASLICEID is determined from the context of the SQL statement that uses it. The database partition number returned on transition variables and tables is derived from the current transition values of the distribution key columns.

Before an unqualified reference to 'DATASLICEID' is translated as a NODENUMBER() function, an attempt is made to resolve the reference to one of the following items:

- A column within the current SQL query
- A local variable
- A routine parameter
- A global variable

Avoid using 'DATASLICEID' as a column name or a variable name while DATASLICEID pseudocolumn is needed. All limitations of the DBPARTITIONNUM function (and its alternative, the NODENUMBER function) apply to the DATASLICEID pseudocolumn.

SQL compatibility enhancements (6 of 7)▪ **Synonyms**

SYNONYM	TO CALL FOLLOWING FUNCTION
COVAR_POP	COVARIANCE
LOG	LN
POW	POWER
RANDOM	RAND
STDDEV_POP	STDDEV
STRPOS	POSSTR
STRLEFT	LEFT
STRRIGHT	RIGHT
VAR_POP	VARIANCE
VAR_SAMP	VARIANCE_SAMP

Any unresolved and unqualified column reference to the DATASLICEID pseudocolumn is converted to NODENUMBER function and returns the database partition number for a row. For example, if DATASLICEID is used in a SELECT clause, the database partition number for each row is returned in the result set.

The specific row (and table) for which the database partition number is returned by DATASLICEID is determined from the context of the SQL statement that uses it. The database partition number returned on transition variables and tables is derived from the current transition values of the distribution key columns.

Before an unqualified reference to 'DATASLICEID' is translated as a NODENUMBER() function, an attempt is made to resolve the reference to one of the following items:

- A column within the current SQL query
- A local variable
- A routine parameter
- A global variable

Avoid using 'DATASLICEID' as a column name or a variable name while DATASLICEID pseudocolumn is needed. All limitations of the DBPARTITIONNUM function (and its alternative, the NODENUMBER function) apply to the DATASLICEID pseudocolumn.

SQL compatibility enhancements (7 of 7)

Compatibility features for Netezza Performance Server (NPS)

- **SET SQL_COMPAT= 'NPS'**
 - global variable activates the following optional NPS features
- **Double-dot notation**
 - When operating in NPS compatibility mode, you can use double-dot notation to specify a database object.
- **TRANSLATE parameter syntax**
 - TRANSLATE (char-string-exp, from-string-exp, to-string-exp)
- **Operators**
 - Operators ^ and ** are both interpreted as the exponential operator
 - Operator # is interpreted as bitwise XOR
- **Grouping by SELECT clause columns**
 - Can specify ordinal position or exposed name of a SELECT clause column when grouping the results of a query.
- **Routines written in NZPLSQL**
 - NZPLSQL language can be used in addition to the SQL PL language.

You can use the SQL_COMPAT global variable to activate the following optional NPS compatibility features.

SQL_COMPAT global variable is new

- Built-in global variable specifies SQL compatibility mode.
- Its value determines which set of syntax rules are applied to SQL queries.
- It is a read/write variable, with values maintained by the user.
- Type VARCHAR(3).
- Schema SYSIBM.
- Scope of global variable is session.
- Values:
 - NULL - Default setting ('DB2') is used.
 - 'DB2' - DB2 syntax rules are applied to SQL queries.
 - 'NPS' - Netezza syntax rules are applied to SQL queries.
 - If used, some SQL behavior will differ from what is documented in SQL reference information. To determine the potential effects of a compatibility feature on your SQL applications, see student notes and "Compatibility Features for Netezza" in Knowledge Center.

LOCALTIMESTAMP special register synonym of CURRENT TIMESTAMP special register.

The NPS value of SQL_COMPAT global variable in DB2 11 provides features that enable applications that were written for a Netezza Performance Server (NPS) database to use a DB2 database without having to be rewritten.

In addition to the Netezza Performance Server (NPS) compatibility features that are always active (see Compatibility features that are always active), you can use the SQL_COMPAT global variable to activate the following optional NPS compatibility features:

- Double-dot notation - When operating in NPS compatibility mode, you can use double-dot notation to specify a database object.
- TRANSLATE parameter syntax - The syntax of the TRANSLATE parameter depends on whether NPS compatibility mode is being used.
- Operators - Which symbols are used to represent operators in expressions depends on whether NPS compatibility mode is being used.
- Grouping by SELECT clause columns - When operating in NPS compatibility mode, you can specify the ordinal position or exposed name of a SELECT clause column when grouping the results of a query.
- Routines written in NZPLSQL - When operating in NPS compatibility mode, the NZPLSQL language can be used in addition to the SQL PL language.

New Built-In Functions

ADD_DAYS	ADD_HOURS	ADD_MINUTES
ADD_SECONDS	ADD_YEARS	AGE
BTRIM	COLLATION_KEY	COVARIANCE_SAMP
CUME_DIST	DATE_PART	DATE_TRUNC
DAYOFMONTH	DAYS_BETWEEN	DAYS_TO_END_OF_MONTH
EXTRACT	FIRST_DAY	HASH
HAS4	HASH8	MINUTES_BETWEEN
MEDIAN	HOURS_BETWEEN	INTNAND,INTNOR,INTNXOR,INTNNOT
NEXT_MONTH	NEXT_QUARTER	NEXT_WEEK
NEXT_YEAR	NOW	PERCENTILE_CONT
RAWTOHEX	PERCENT_RANK	PERCENTILE_DISC
THIS_MONTH	STDDEV_SAMP	SECONDS_BETWEEN
THIS_QUARTER	THIS_YEAR	VARIANCE_SAMP
THIS_WEEK	TO_HEX	WEEKS_BETWEEN
WIDTH_BUCKET	YMDBETWEEN	YEARSBETWEEN

The above built-in functions are new in 11.

HEXTORAW scalar function returns a different data type than previous releases.

Discontinued functionality in 11.1

Discontinued Capability	Resolution
Cubing Services	Migrate to Cognos Dynamic Cubes.
Raw logs - use of raw devices for database logging	Using file system with non-buffered I/O capabilities enabled, such as Concurrent I/O (CIO) or Direct I/O (DIO), yields performance comparable to raw devices.

- Use of raw devices for database logging deprecated since DB2 9.1 and discontinued in DB2 11.1.
- Change **newlogpath** to file system directory:
**db2 UPDATE DATABASE CONFIGURATION
USING newlogpath /disk2/newlogdir**
- Change does not become effective until database in consistent state and all users are disconnected from database.
- Database manager moves logs to new location after database activated implicitly or explicitly.



Use of raw devices for database logging has been deprecated since DB2 9.1 and is discontinued in DB2 11.1. Using file system with non-buffered I/O capabilities enabled, such as Concurrent I/O (CIO) or Direct I/O (DIO), yields performance comparable to raw devices.

Deprecated functionality in 11.1

Deprecated Capability	Resolution
Private Sort Memory: SHEAPTHES in DBM CFG. Not compatible with STMM or column organized tables.	Use shared sort memory instead: SHEAPTHRES_SHR in DB CFG.
Intelligent Mining	Currently no replacement for this feature.
Text Analytics	Currently no replacement for this feature.
Resource Description Framework (RDF) application development	Currently no replacement for this feature.

Summary of deprecated functionality in DB2 11.1.

Deprecated APIs, Commands, command parameters, monitor elements, and statements

Deprecated Capability	Resolution
-disk sub-option under <code>db2cluster -cfs -list -filesystem <fsname></code>	Info already displayed without option.
db2top command	Use dsmtop - DB2 text-based monitoring tool command
SET PASSWORD statement	For data encryption, see DB2 native encryption, and for column masking, see RCAC, in Knowledge Center.
GETHINT scalar function	For data encryption, see DB2 native encryption, and for column masking, see RCAC, in Knowledge Center.
ENCRYPT scalar function	For data encryption, see DB2 native encryption, and for column masking, see RCAC, in Knowledge Center.
DECRYPT_BIN scalar function	For data encryption, see DB2 native encryption, and for column masking, see RCAC, in Knowledge Center.
DECRYPT_CHAR scalar function	For data encryption, see DB2 native encryption, and for column masking, see RCAC, in Knowledge Center.

Summary of deprecated APIs, Commands, command parameters, monitor elements and statements in DB2 11.1.

Changes to System Catalog Views

- Most modifications to catalog views consist of new columns, changed descriptions, changed column data types, and increased column lengths.

- SYSCAT.COLUMNS
- SYSCAT.DATAPARTITIONS
- SYSCAT.MEMBERSUBSETMEMBERS
- SYSCAT.ROLEAUTH
- SYSCAT.ROUTINES
- SYSCAT.SCHEMAAUTH
- SYSCAT.TABLES
- SYSCAT.WORKLOADAUTH
- SYSSTAT.COLUMNS
- SYSSTAT.TABLES



The following system catalog views changed in 11.1. Most modifications to catalog views consist of new columns, changed descriptions, changed column data types, and increased column lengths.

New MON_GET_LATCH table function

- Returns list of all latches in current member.

```
SELECT SUBSTR(LATCH_NAME,1,40) LATCH_NAME, SUBSTR(MEMORY_ADDRESS,1,20) ADDRESS, EDU_ID,
       SUBSTR(EDU_NAME,1,20) EDU_NAME, APPLICATION_HANDLE, MEMBER, LATCH_STATUS, LATCH_WAIT_TIME
FROM TABLE ( MON_GET_LATCH( CLOB('<LATCH_STATUS>C</LATCH_STATUS>
                               <LATCH_NAME>SQLQ_LT_SQLB_PTBL__POOL_TABLE_LATCH</LATCH_NAME>', -2) )
ORDER BY LATCH_NAME, LATCH_STATUS
```

LATCH_NAME	ADDRESS	EDU_ID	...
SQLQ_LT_SQLB_PTBL__pool_table_latch	0x70000004108B910	37911	...
SQLQ_LT_SQLB_PTBL__pool_table_latch	0x70000004108B910	37654	...
SQLQ_LT_SQLB_PTBL__pool_table_latch	0x70000004108B910	37140	...

Output for query continued:

... EDU_NAME	APPLICATION_HANDLE	MEMBER	LATCH_STATUS	LATCH_WAIT_TIME
... db2agent (SAMPLE)	124	0	H	-
... db2agent (SAMPLE)	121	0	W	74956
... db2agent (SAMPLE)	119	0	W	55446

3 record(s) selected.

- See Knowledge Center for search_args parameter details on **application_handle**, **latch_name**, **edu_id**, and **latch_status**.

One of the following authorizations is required:

- EXECUTE privilege on the routine
- DATAACCESS authority
- DBADM authority
- SQLADM authority

The schema is SYSPROC.

Built-in administrative views and routine changes (1 of 3)

NAME	DESCRIPTION
ADMINEMPTYTABLES, ADMIN_GET_TEMP_TABLES	Retrieve information for temporary tables
ADMIN_GET_ENCRYPTION_INFO	Get database encryption settings
ADMIN_MOVE_TABLE	Move tables online
ADMIN_ROTATE_MASTER_KEY	Change the database master key
MON_FORMAT_LOCK_NAME	Format the internal lock name and return details
MON_FORMAT_XML_COMPONENT_ TIMES_BY_ROW	Get formatted row-based component times
MON_FORMAT_XML_METRICS_BY_ROW	Get formatted row-based output for all metrics
MON_FORMAT_XML_TIMES_BY_ROW	Get formatted row-based combined hierarchy wait and processing times
MON_GET_ACTIVITY	Return a list of activities
MON_GET_ACTIVITY_DETAILS	Return information about an activity as an XML document

The above administrative views and routines changed in 11.1. Most modifications consist of new columns, new values, changed column data types, and increased column lengths.

Built-in administrative views and routine changes (2 of 3)

NAME	DESCRIPTION
MON_GET_AGENT	List agents running in a service class
MON_GET_CONNECTION	Get connection metrics
MON_GET_CONNECTION_DETAILS	Get connection metrics as an XML document
MON_GET_DATABASE	Get database metrics
MON_GET_DATABASE_DETAILS	Get database metrics as an XML document
MON_GET_EXTENDED_LATCH_WAIT	Return information for latches
MON_GET_HADR	Returns high availability disaster recovery (HADR) monitoring information
MON_GET_PKG_CACHE_STMT	Get package cache statement metrics
MON_GET_PKG_CACHE_STMT_DETAILS	Get package cache statement metrics as an XML document
MON_GET_ROUTINE	Get aggregated routine execution metrics
MON_GET_ROUTINE_DETAILS	Get aggregated routine execution metrics as an XML document

The above administrative views and routines changed in 11.1. Most modifications consist of new columns, new values, changed column data types, and increased column lengths.

Built-in administrative views and routine changes (3 of 3)

NAME	DESCRIPTION
MON_GET_RTS_RQST	Get service subclass metrics
MON_GET_SERVICE_SUBCLASS	Get service subclass metrics as an XML document
MON_GET_SERVICE_SUBCLASS_DETAILS	Get connection metrics as an XML document
MON_GET_TABLE	Get table metrics
MON_GET_TABLE_USAGE_LIST	Returns information from a table usage list
MON_GET_UNIT_OF_WORK	Get unit of work metrics
MON_GET_UTILITY	Get utilities running on the database
MON_GET_WORKLOAD	Get workload metrics
MON_GET_WORKLOAD_DETAILS	Get workload metrics as an XML document
MON_GET_UNIT_OF_WORK_DETAILS	Get unit of work metrics as XML document
WLM_ALTER_MEMBER_SUBSET	Procedure alters a member subset object
WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES	Return a list of activities

The above administrative views and routines changed in 11.1. Most modifications consist of new columns, new values, changed column data types, and increased column lengths.

- Course Title: **DB2 11 New Features and Database Migration**
Course Code: **CL315** (classroom)



Course number may vary depending upon language and provider.

- This course can be customized to meet your needs.
- Watch for new DB2 11.1 Certification Exams.

Course Code CL315 provides training on DB2 11 New Features and Database Migration.

DB2 11.1 LUW CERTIFICATION OPPORTUNITIES

- IBM Certified Database Associate
 - Test C2090-599: IBM DB2 11.1 Fundamentals for LUW
- IBM Certified Database Administrator
 - Test C2090-600: IBM DB2 11.1 DBA for LUW
- IBM Certified Advanced Database Administrator
 - Test C2090-601: IBM DB2 11.1 Advanced DBA for LUW

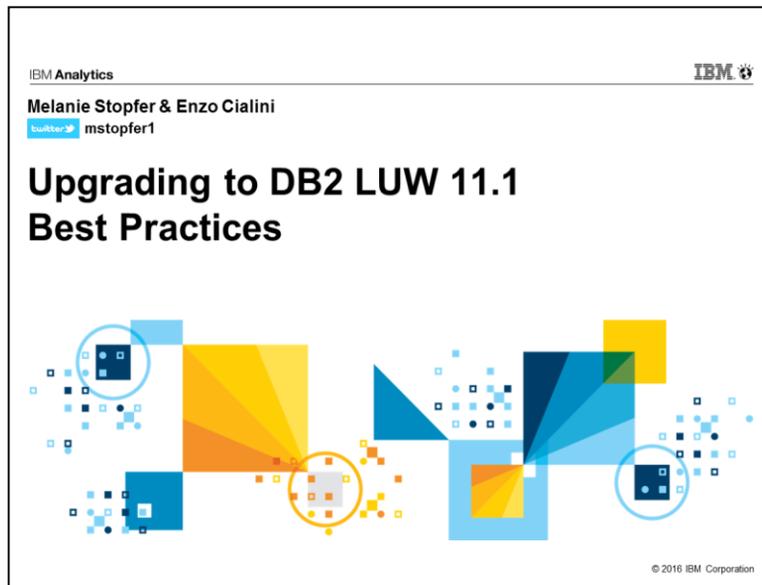
UNDER CONSTRUCTION
CONTENT WILL BE AVAILABLE SOON



* Not available yet.

* Watch http://www-03.ibm.com/certify/certs/index_unit_an.shtml

DB2 11.1 certification examinations are being written.



Melanie Stopfer is a IBM Consulting Learning Specialist and Developer and recognized worldwide as a DB2, IBM BLU and Analytics subject matter expert. She has provided in-depth technical customer support in recovery, performance and database upgrade best practices since 1988. In 2009, Melanie was the first DB2 LUW speaker to be inducted into the IDUG Speaker Hall of Fame, selected Best Overall Speaker a record breaking seven times (IDUG EMEA 2009, 2011, 2012, 2015; IDUG NA 2009, 2012, 2014), and included in Top 10 Overall Speakers in all years presenting. She has successfully presented at all IBM IOD and Insight conferences and been recognized as a top presenter. Please connect with Melanie on Twitter at @mstopfer1 and LinkedIn.

Thank you to Enzo Cialini for formatting my previous presentation into the IBM Analytics format, starting the upgrade, and motivating me to complete the DB2 11.1 Upgrade Best Practices presentation.