

SAP NetWeaver '04



SAP[®] BW Administration
Tasks in Multi-Partition
Installations:

IBM DB2 Universal Database
for UNIX and Windows

Document Version 1.00 – April 30th, 2004



SAP AG
Neurottstraße 16
69190 Walldorf
Germany
T +49/18 05/34 34 24
F +49/18 05/34 34 20
www.sap.com

© Copyright 2004 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C[®], World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

Disclaimer

Some components of this product are based on Java[™]. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressly prohibited, as is any decompilation of these components.

Any Java[™] Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

Documentation in the SAP Service Marketplace

You can find this documentation at the following Internet address:
service.sap.com/instguides

Terms for Included Open Source Software

This SAP software contains also the third party open source software products listed below. Please note that for these third party products the following special terms and conditions shall apply.

SAP License Agreement for STLport

SAP License Agreement for STLport
between

SAP Aktiengesellschaft
Systems, Applications, Products in Data Processing
Neurottstrasse 16
69190 Walldorf
Germany
(hereinafter: SAP)

and

you
(hereinafter: Customer)

1. Subject Matter of the Agreement

- a. SAP grants Customer a non-exclusive, non-transferable, royalty-free license to use the STLport.org C++ library (STLport) and its documentation without fee.
- b. By downloading, using, or copying STLport or any portion thereof Customer agrees to abide by the intellectual property laws, and to all of the terms and conditions of this Agreement.
- c. The Customer may distribute binaries compiled with STLport (whether original or modified) without any royalties or restrictions.
- d. Customer shall maintain the following copyright and permission notices on STLport sources and its documentation unchanged:
Copyright 2001 SAP AG

- e. The Customer may distribute original or modified STLport sources, provided that:
 - The conditions indicated in the above permission notice are met;
 - The following copyright notices are retained when present, and conditions provided in accompanying permission notices are met:

Copyright 1994 Hewlett-Packard Company
Copyright 1996,97 Silicon Graphics Computer Systems, Inc.
Copyright 1997 Moscow Center for SPARC Technology.
Copyright 1999,2000 Boris Fomitchev
Copyright 2001 SAP AG

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Hewlett-Packard Company makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Silicon Graphics makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Moscow Center for SPARC

Technology makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Boris Fomitchev makes no representations about the suitability of this software for any purpose. This material is provided "as is", with absolutely no warranty expressed or implied. Any use is at your own risk. Permission to use or copy this software for any purpose is hereby granted without fee, provided the above notices are retained on all copies. Permission to modify the code and to distribute modified code is granted, provided the above notices are retained, and a notice that the code was modified is included with the above copyright notice.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. SAP makes no representations about the suitability of this software for any purpose. It is provided with a limited warranty and liability as set forth in the License Agreement distributed with this copy. SAP offers this liability and warranty obligations only towards its customers and only referring to its modifications.

2. Support and Maintenance

SAP does not provide software maintenance for the STLport. Software maintenance of the STLport therefore shall be not included.

All other services shall be charged according to the rates for services quoted in the SAP List of Prices and Conditions and shall be subject to a separate contract.

3. Exclusion of warranty

As the STLport is transferred to the Customer on a loan basis and free of charge, SAP cannot guarantee that the STLport is error-free, without material defects or suitable for a specific application under third-party rights. Technical data, sales brochures, advertising text and quality descriptions produced by SAP do not indicate any assurance of particular attributes.

4. Limited Liability

- a. Irrespective of the legal reasons, SAP shall only be liable for damage, including unauthorized operation, if this (i) can be compensated under the Product Liability Act or (ii) if caused due to gross negligence or intent by SAP or (iii) if based on the failure of a guaranteed attribute.
- b. If SAP is liable for gross negligence or intent caused by employees who are neither agents or managerial employees of SAP, the total liability for such damage and a maximum limit on the scope of any such damage shall depend on the extent to which its occurrence ought to have anticipated by SAP when concluding the contract, due to the circumstances known to it at that point in time representing a typical transfer of the software.
- c. In the case of Art. 4.2 above, SAP shall not be liable for indirect damage, consequential damage caused by a defect or lost profit.
- d. SAP and the Customer agree that the typical foreseeable extent of damage shall under no circumstances exceed EUR 5,000.
- e. The Customer shall take adequate measures for the protection of data and programs, in particular by making backup copies at the minimum intervals recommended by SAP. SAP shall not be liable for the loss of data and its recovery, notwithstanding the other limitations of the present Art. 4 if this loss could have been avoided by observing this obligation.
- f. The exclusion or the limitation of claims in accordance with the present Art. 4 includes claims against employees or agents of SAP.

Typographic Conventions

Type Style	Description
<i>Example Text</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
EXAMPLE TEXT	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
Example text	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
Example text	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE TEXT	Keys on the keyboard, for example, F2 or ENTER.

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see *Help on Help → General Information Classes and Information Classes for Business Information Warehouse* on the first page of any version of *SAP Library*.

Contents

SAP BW Administration Tasks in Multi-Partition Installations: IBM DB2 Universal Database for UNIX and Windows:	8
1 Naming Conventions	9
2 Standard SAP BW System 3.5 on DB2 UDB for UNIX and Windows	10
2.1 Default Partition Layout	10
2.2 Default Database and File System Layout (UNIX)	14
2.3 Default Database and Directory Layout (Windows)	15
2.4 Usage of the Default BW Dimension, Fact and ODS and PSA Tablespaces	16
3 Database Layout Planning	17
3.1 Number of Partitions	17
3.2 Data Distribution of InfoCubes	18
3.3 Data Distribution of ODS Objects	20
3.4 Data Distribution of PSA Tables	21
3.5 Partition Containing the Database Catalog	22
3.6 Scalable Partition Layout	23
4 Creation of Large Infocubes, ODS Objects, and PSA Tables	25
4.1 Creating Database Partition Groups	25
4.1.1 Creating a Database Partition Group for InfoCubes	26
4.1.2 Creating Database Partition Groups for ODS Objects	27
4.1.3 Creating Database Partition Groups for PSA Tables	27
4.2 Creating Buffer Pools	28
4.3 Creating Tablespaces	29
4.4 Creating Data Classes	33
4.4.1 Assigning InfoCubes and Their Aggregates to Data Classes	34
4.4.2 Assigning ODS Objects to Data Classes	35
4.4.3 Assigning PSA Tables to Data Classes	35
5 Adding of New Partitions to the Database	36
5.1 Preparing New Partitions	37
5.1.1 Creating Additional Partitions	37
5.1.2 Updating the Database Configuration for Each New Partition	39
5.1.3 Adding Containers to Temporary Tablespaces on Each New Partition	41
5.2 Redistributing the Data with DB6CONV	42
5.2.1 Creating New Database Partition Groups	42
5.2.2 Creating New Tablespaces in the New Database Partition Group	43
5.2.3 Checking the Partitioning Keys of All Partitioned Tables	43
5.2.4 Redistributing All Partitioned Tables	44
5.3 Redistributing the Data with DB2 REDISTRIBUTE	46
5.3.1 Adding Partitions to Database Partitions Groups	46
5.3.2 Creating Tablespace Containers on Each New Partition	47

5.3.3 Checking The Partitioning Keys of all Partitioned Tables.....	47
5.3.4 Redistributing Database Partition Groups	48
6 Best Practices	49
6.1 Data Clustering of InfoCube Fact Tables	49
6.2 Controlling Database Log Space Consumption	51
6.3 Dropping Indexes Before Deletion and Compression of InfoPackages	52
6.4 Increasing the Buffer Pool Hit Ratio	53
6.5 Using Temporary Tablespaces	56
7 Additional Sources of Information	57

SAP BW 3.5 Administration Tasks on IBM DB2 Universal Database for UNIX and Windows

Purpose

This documentation describes the physical database layout, the administration tasks that you have to perform and performance recommendations for an SAP BW 3.5 system if your database is IBM DB2 Universal Database for UNIX or Windows.

It focuses especially on an environment with multiple database partitions and is also valid for SAP OLAP (online analytical processing) systems that are based on SAP BW 3.5. SAP OLAP systems are, for example, SAP BW, SAP SCM, SAP SRM (formerly known as SAP APO) and SAP SEM.

The documentation contains the following information:

- [Naming conventions \[Page 9\]](#)
- A description of the [standard database layout \[Page 10\]](#)
- Information about and recommendations for the [database layout planning \[Page 17\]](#)
- Information about how to [create large InfoCubes, ODS Objects and PSA tables \[Page 25\]](#)
- Information about how to [add new partitions to the database \[Page 36\]](#)
- [Best practices \[Page 49\]](#) with information about special database-related recommendations
- [Additional sources of information about SAP BW \[Page 57\]](#)



The information provided in this documentation mainly applies to **newly installed SAP BW 3.5 systems**.

For additional information about administration tasks in SAP BW multi-partition installations, see **SAP Note 688647**.

Constraints

When using this documentation, you have to consider the following information:

- SAP BW 3.5 is **only** available for **version 8** of IBM DB2 Universal Database for UNIX and Windows.
- Most of the recommendations in this documentation are also valid for earlier SAP BW releases, for example, SAP BW 3.1 or lower.

There are differences in the database layout between the different SAP BW releases, which are **not** explicitly described in this documentation. Make sure that you adapt the given recommendation of this documentation to the requirements of your SAP BW release, for example:

- ODS and PSA tablespaces are **not** distributed across multiple partitions with SAP BW 2.x.
- For SAP BW 3.5, a page size of 16 KB is used for fact tablespace. For SAP BW release 2.x, a page size of 8 KB is used.

1 Naming Conventions

In this documentation, the following naming conventions apply:

Database Versions

Formal Name	Short Name
<i>IBM DB2 Universal Database for UNIX and Windows</i>	<i>DB2 UDB for UNIX and Windows</i>
<i>IBM DB2 Universal Database Enterprise Server Edition for UNIX and Windows</i>	<i>DB2 UDB ESE for UNIX and Windows</i>

Database Terminology

Term	Description
<i>Partition or database partition</i>	Part of the database that consists of its own data, indexes, configuration files and transaction logs
<i>Database partition group</i>	Group of one or more partitions

New Tablespace Names

As of SAP BW 3.1 Content the tablespace names have changed. Except for temporary tablespaces, the prefix <SAPSID># replaces the former prefix PSAP. Throughout this documentation the prefix <SAPSID># is used.



If your system is a SAP BW system with a release lower than 3.1 Content, the tablespace names do not change when you perform an upgrade to SAP BW 3.5. The prefix PSAP is still used.

2 Standard SAP BW System 3.5 on DB2 UDB for UNIX and Windows

The following sections provide information about:

- [Default Partition Layout \[Page 10\]](#)
- [Default Database and File System Layout \(UNIX\) \[Page 14\]](#)
- [Default Database and Directory Layout \(Windows\) \[Page 15\]](#)
- [Usage of the Default BW Dimension, Fact and ODS Tablespaces \[Page 16\]](#)

2.1 Default Partition Layout

This section provides information about default buffer pools, database partition groups and tablespaces that are created during an SAP NetWeaver installation. Throughout this section, we distinguish between SAP base database objects and BW-specific database objects.



Make sure that you apply all SAP BW-specific steps of an SAP NetWeaver installation as described in the documentation *SAP NetWeaver 04 Installation Guide: SAP Web Application Server ABAP 6.40 on <Operating system platform>: IBM DB2 Universal Database for UNIX and Windows*. You can find this documentation in SAP Service Marketplace at service.sap.com/instguidesNW04 → *Installation*.

Buffer Pools

The following default buffer pools are created during an SAP system installation:

- In a **non-Unicode** installation, default buffer pool `IBMDEFAULTTBP` is automatically created and exists on all partitions. `IBMDEFAULTTBP` has a page size of 4 KB.
- In a **Unicode** installation, the default buffer pools `IBMDEFAULTTBP` and `BP_STD_16K` are automatically created. Both buffer pools exist on all partitions.
`IBMDEFAULTTBP` has a page size of 4 KB and `BP_STD_16K` has a page size of 16 KB.
- You manually create buffer pool `BP_BW_16K` during an SAP NetWeaver installation. `BP_BW_16K` exists on all partition and has a page size of 16 KB.

The following tables provide information about the recommended initial buffer pool sizes for non-Unicode and Unicode SAP BW installations.



The buffer pool for aggregate data is **not** part of the SAP NetWeaver installation and must be created manually later on.

• **Non-Unicode Installation**

Buffer Pool Used for	Buffer Pool Name	Page Size	Number of Pages
Dimension data, SAP BW master data and SAP base data	IBMDEFAULTBP	4 KB	14000 pages on partition 0, 1000 pages on all other partitions
Fact, ODS or PSA data	BP_BW_16K	16 KB	Roughly one quarter of real memory available for one database partition. BP_BW_16K is created on all partitions.  If your real memory for one database partition is 2 GB, 500 MB (that is, 31250 pages) are sufficient.
Aggregate data	User-defined	16 KB	7500 pages

• **Unicode Installation**

Buffer Pool Used For	Buffer Pool Name	Page Size	Number of Pages
DB2 catalog data	IBMDEFAULTBP	4 KB	1000 pages on all partitions
Dimension data, SAP BW master data and SAP base data	BP_STD_16K	16 KB	6000 pages on partition 0, 1000 pages on all other partitions

Fact, ODS, PSA data	BP_BW_16K	16 KB	<p>Roughly one quarter of real memory available for one database partition. BP_BW_16K is created on all partitions.</p> <p> If your real memory for one database partition is 2 GB, 500 MB (that is, 31250 pages) are sufficient.</p>
Aggregate data	User-defined	16 KB	7500 pages

Database Partition Groups

SAP Base Database Partition Group

The SAP base database partition group is SAPNODEGRP_<SAPSID> and contains partition 0.

BW-Specific Database Partition Groups

The following BW-specific default database partition groups are created during the SAP system installation:

- The default dimension database partition group NGRP_DIM_<SAPSID> that contains partition 0
- The default fact database partition group NGRP_FACT_<SAPSID> that contains all partitions
- The PSA and ODS database partition group NGRP_ODS_<SAPSID> that contains all partitions

Tablespaces

Base Tablespaces

The following information applies to all SAP base standard tablespaces in a non-Unicode or Unicode installation:

- **Non-Unicode** installation
 - All base tablespaces are on partition 0 and use the IBMDEFAULTBP buffer pool and have a page size of 4 KB.
 - The IBMDEFAULTBP buffer pool exists on all partitions and has a page size of 4 KB.
 - The temporary tablespace PSAPTEMP uses the buffer pool IBMDEFAULTBP and the temporary tablespace PSAPTEMP16 uses the buffer pool BP_BW_16K.

- **Unicode installation**

- All base tablespaces are on partition 0 and use the BP_STD_16K buffer pool and have a page size of 16 KB.
- The BP_STD_16K buffer pool exists on all partitions and has a page size of 16 KB.
- The temporary tablespace PSAPTEMP uses the IBMDEFAULTBP buffer pool and the temporary tablespace PSAPTEMP16 uses the BP_BW_16K buffer pool.



If you are upgrading a system, the tablespace names **do not** change. If the source system release is lower than SAP BW 3.1 Content, the prefix PSAP remains.

BW-Specific Tablespaces

The following BW-specific default tablespaces are created during the SAP system installation:

- The **dimension tablespaces** <SAPSID>#DIMD (data tablespace) and <SAPSID>#DIMI (index tablespace) are created in the database partition group NGRP_DIM_<SAPSID>. The corresponding data class is DDIM.
 - In a **non-Unicode** installation, the dimension tablespaces have a page size of 4 KB and are assigned to the IBMDEFAULTBP buffer pool.
 - In a **Unicode** installation, the dimension tablespaces have a page size of 16 KB and are assigned to the BP_STD_16K buffer pool.



Both buffer pools are also used for SAP BW master data and SAP base tablespaces.

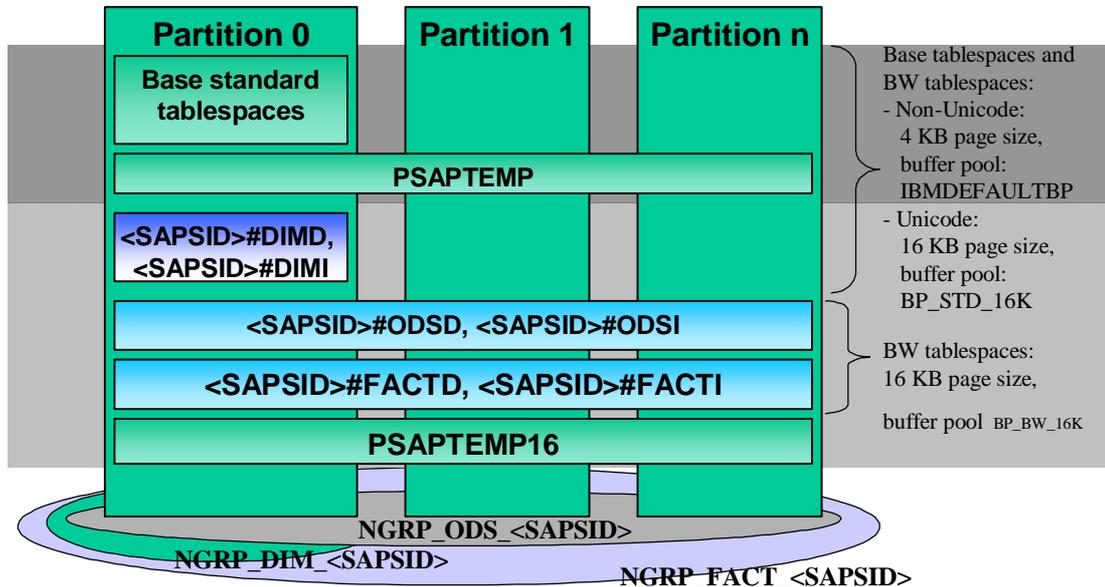
- The **fact tablespaces** <SAPSID>#FACTD (data tablespace) and <SAPSID>#FACTI (index tablespace) are created in the database partition group NGRP_FACT_<SAPSID>. The corresponding data class is DFACT. The fact tablespaces have a page size of 16 KB and are assigned to the BP_BW_16K buffer pool.
- The **tablespaces for PSA and ODS tables** are <SAPSID>#ODSD (data tablespace) and <SAPSID>#ODSI (index tablespace) and are created in the database partition group NGRP_ODS_<SAPSID>. The corresponding data class is DODS. The PSA and ODS tablespaces have a page size of 16 KB and are assigned to the BP_BW_16K buffer pool.



If you are upgrading a system, the tablespace names **do not** change. If the source system release was lower than SAP BW 3.1 Content, the prefix PSAP remains.

The following graphic shows the SAP BW system tablespaces after an SAP BW server installation as part of the SAP NetWeaver '04 installation.

SAP BW Tablespace Layout After the Installation



2.2 Default Database and File System Layout (UNIX)

The base tablespace containers are located in:

```
/db2/<SAPSID>/sapdata1/NODE0000/...
...
/db2/<SAPSID>/sapdata6/NODE0000/...
```

The temporary tablespace containers (SMS) are located in:

```
/db2/<SAPSID>/saptemp1/NODE<NNNN>/temp4
/db2/<SAPSID>/saptemp1/NODE<NNNN>/temp16
```



SMS tablespaces are growing and shrinking dynamically. Therefore, the 4 KB and 16 KB temporary tablespaces should be located in the same file system.

The default dimension tablespace containers (DMS) are:

```
/db2/<SAPSID>/sapdata7/NODE0000/<SAPSID>#DIMD.container001
/db2/<SAPSID>/sapdata7/NODE0000/<SAPSID>#DIMI.container001
```

The default fact tablespace containers (DMS) are:

/db2/<SAPSID>/sapdata8/NODE<NNNN>/<SAPSID>#FACTD.container001

/db2/<SAPSID>/sapdata8/NODE<NNNN>/<SAPSID>#FACTI.container001

The PSA/ODS tablespace containers (DMS) are:

/db2/<SAPSID>/sapdata9/NODE<NNNN>/<SAPSID>#ODSD.container001

/db2/<SAPSID>/sapdata9/NODE<NNNN>/<SAPSID>#ODSI.container001

2.3 Default Database and Directory Layout (Windows)

The base tablespace containers are located in:

<drive>:\db2\<SAPSID>\sapdata1\NODE0000\...

...

<drive>:\db2\<SAPSID>\sapdata6\NODE0000\...

The temporary tablespace containers (SMS) are located in:

<drive>:\db2\<SAPSID>\saptemp1\NODE<NNNN>\temp4

<drive>:\db2\<SAPSID>\saptemp1\NODE<NNNN>\temp16



SMS tablespaces are growing and shrinking dynamically. Therefore, the 4 KB and 16 KB temporary tablespaces should be located in the same file system.

The default dimension tablespace containers (DMS) are:

<drive>:\db2\<SAPSID>\sapdata7\NODE0000\<SAPSID>#DIMD.container000

<drive>:\db2\<SAPSID>\sapdata7\NODE0000\<SAPSID>#DIMI.container000

The default fact tablespace containers (DMS) are:

<drive>:\db2\<SAPSID>\sapdata8\NODE<NNNN>\<SAPSID>#FACTD.container000

<drive>:\db2\<SAPSID>\sapdata8\NODE<NNNN>\<SAPSID>#FACTI.container000

The PSA and ODS tablespace containers (DMS) are:

<drive>:\db2\<SAPSID>\sapdata9\NODE<NNNN>\<SAPSID>#ODSD.container000

<drive>:\db2\<SAPSID>\sapdata9\NODE<NNNN>\<SAPSID>#ODSI.container000

2.4 Usage of the Default BW Dimension, Fact and ODS and PSA Tablespaces

The SAP Web AS 6.40 ABAP installation package contains a BW-specific database layout. You can use this layout as an example of setting up an SAP BW system in a test environment. You can create first InfoCubes, ODS objects and PSA tables. This will help you to learn how to administrate an SAP BW system and is also useful for testing purposes.



The default database layout **cannot** be used as a base for a production BW system.

The following sections describe how to plan and set up a productive environment.

3 Database Layout Planning

Purpose

The layout of your database has a large impact on the performance of your SAP BW system. It therefore requires careful planning.

Process Flow

1. You check and decide how much hardware you will need. The hardware sizing must reflect the size of the final hardware needed for the installation as closely as possible.

For general hardware sizing information for SAP BW, see SAP Service Marketplace at service.sap.com/bw → *Performance*.

To make a first sizing estimation, we recommend that you use the Web-based tool SAP Quick Sizer that is available on SAP Service Marketplace at service.sap.com/quicksizer.

You determine the number of machines and the amount of processors, memory, and disks that you need for the chosen hardware platform.
2. You need to plan the database layout itself.

To be able to correctly plan the database layout, you must take the following sections into consideration:
 - [Number of Database Partitions \[Page 17\]](#)
 - [Data Distribution of InfoCubes \[Page 18\]](#)
 - [Data Distribution of ODS Objects \[Page 20\]](#)
 - [Data Distribution of PSA Tables \[Page 21\]](#)
 - [Partition containing the Database Catalog \[Page 22\]](#)
 - [Scalable Partition Layout \[Page 23\]](#)
3. You [create large InfoCubes, ODS objects and PSA tables \[Page 25\]](#).

3.1 Number of Partitions

Multiple partitions have several advantages for large databases that have more than 300 GB, for example:

- The administration tasks **backup/restore** and **reorg/runstats** benefit from the inter-partition parallelism and run faster.
- Since the queries are processed in parallel, the query performance for large tables improves.



Intra-partition parallelism splits an SQL operation into independent sub-operations that can be run in parallel on a number of CPUs. This can be done in a single-partition database or separately in each partition of a multi-partition database.

Inter-partition parallelism executes an SQL operation in parallel on each partition of a multi-partition database and accumulates the individual results from all partitions to one, final result. The partitions contain disjoint subsets of the data and an SQL operation is executed on these subsets before the final result is accumulated (also known as function-shipping).

The number of processors on each database partition server determines the number of logical partitions on that machine. We recommend that you have one to four processors per partition.

A good compromise is to have two processors per database partition. This allows a full CPU utilization for single running BW queries if intra-partition parallelism is switched on. In this case, the database manager configuration parameters `INTRA_PARALLEL` is set to `YES`, and `MAX_QUERYDEGREE` is set to 2.

In addition, you can reduce the CPU workload by switching off intra-partition parallelism if your database server reaches CPU utilization limits due to increasing workload (more data, more complex queries, more users).

If you know the number of database partition servers, which you plan to add at a later point in time, you can configure more partitions on the initial database partition servers to avoid data repartitioning. For more information, see [Scalable Partition Layout \[Page 23\]](#).



Make sure that you have enough main memory. Calculate a minimum of 2 GB real memory for each database partition. In general, DB2 scales linear to the number of partitions.

3.2 Data Distribution of InfoCubes

InfoCubes consist of three different types of tables: Dimension tables, fact tables and aggregate tables. Typically, only fact tables grow large in size and should be distributed across multiple partitions.

When planning the data distribution of **InfoCubes**, you have to take the following information into consideration:

- Do **not** distribute dimension tables across **multiple partitions**. Dimension tables should reside on the **same single partition** as the master data tables.
- Since dimension tables are relatively small, the dimension tables of all InfoCubes can be created in the standard dimension tablespace.

If you expect many distinct values in a dimension, you should consider using a line item dimension for this dimension in the InfoCube's layout.

- For all InfoCubes with small fact tables (smaller than 10 GB), the fact tables can be located on a single partition. You create the fact tables of all small InfoCubes in one tablespace. If you have a large number of small InfoCubes, you benefit from parallelism by creating one separate, non-distributed tablespace on each partition and by creating the small fact tables alternately in these separated tablespaces.
- For all InfoCubes with large fact tables (larger than 10 GB), the fact tables should be distributed across multiple partitions.



We recommend that you create a separate database partition group for each large InfoCube and a separate tablespace for the fact tables of this InfoCube. This helps to plan and to administer the space of these large fact tables more individually and to create a separate buffer pool if you require high performance on this InfoCube. For information about the creation of individual database partition groups and tablespaces, see [Creation of Large InfoCubes, ODS Objects, and PSA Tables \[Page 25\]](#).

As **aggregates** are part of an InfoCube, you should also consider the following when planning the data distribution:

- Aggregates are small InfoCubes that contain condensed information from the original InfoCube. Typically, the fact tables of aggregates are small and do not vary in size. The average query response time is in a subsecond time frame.



Do **not** distribute the fact tables of those small aggregates since the communication cost will increase the response time of queries on the aggregates.

- You should store all small aggregates of an InfoCube on a single partition. Similar to small InfoCubes, you can create aggregates of different InfoCubes in one tablespace. If you have a large number of aggregates, you can benefit from parallelism by creating one separate, non-distributed tablespace on each partition and by creating aggregates alternately in these separated tablespaces.

- By default, aggregates are created in the same tablespace as the fact tables.

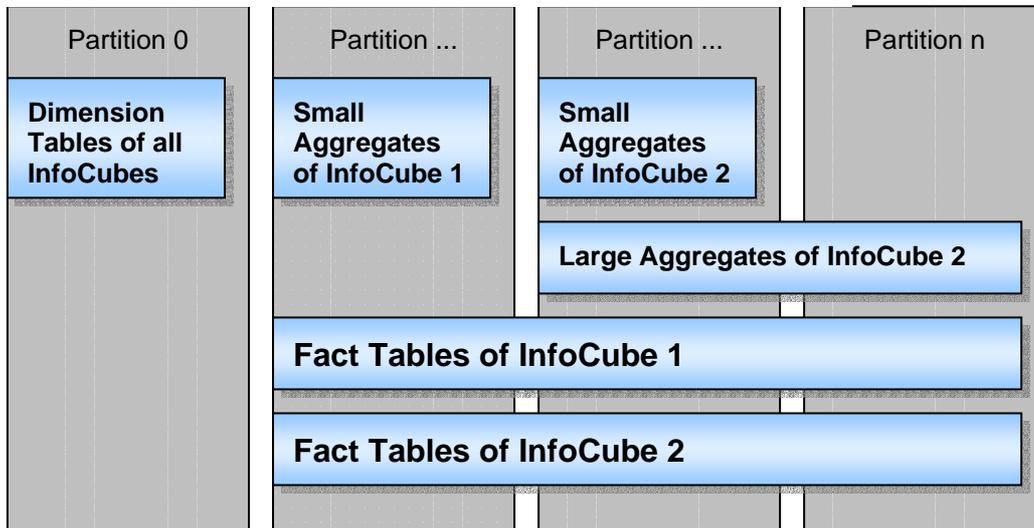
We recommend that you create a separate database partition group and tablespace for aggregate tables. You can create aggregates of different InfoCubes in the same tablespace.

- In certain cases, when you have large aggregates (with single aggregates larger than 10 % of the InfoCube), you should consider distributing the aggregate tablespace.

We recommend that you use a separate tablespace for aggregates. This option allows you to split buffer pools for fact tables and aggregate tables.

The following graphic shows an example of how you can distribute data of two large InfoCubes:

Data Distribution of InfoCubes



The small aggregates of the two InfoCubes shown in the graphic above reside in separated tablespaces on different partitions.

3.3 Data Distribution of ODS Objects

ODS objects consist of three different types of tables: Active tables, activation queue tables and change log tables. Typically, the active table and the change log table grow large in size and should be distributed across multiple partitions.

When planning the data distribution of ODS objects, you have to take the following information into consideration:

- Activation queue tables do **not** contain much data and are emptied when the data is activated. Therefore, activation queue tables should be located on a **single partition**.
- By default, all tables of an ODS object are created in the same tablespace. You should create a separate tablespace for activation queue tables. The reason is that during ODS activation data is read from the activation queue table and is inserted into the active table.

You can create the activation queue tables of all ODS objects in the same tablespace. You should create the active table and the change log table of an ODS object in the **same** tablespace.

- Small ODS objects, which have an active table with less than 10 GB, can be stored on a single partition. You should create the active table and the change log table of all small ODS objects in the **same** tablespace.

- Large ODS objects, which have an active table greater than 10 GB, should be distributed across multiple partitions.



We recommend that you create a separate database partition group for each large ODS object and a separate tablespace for the active table and the change log table of this ODS object. This helps you to plan and to administer the space of these large ODS objects more individually and to create a separate buffer pool. For information about the creation of individual database partition groups and tablespaces, see [Creation of Large InfoCubes, ODS Objects, and PSA Tables \[Page 25\]](#).

- Production ODS objects and PSA tables should be located in different tablespaces and not in the default ODS tablespace.



If you are using the SAP BW default settings, ODS and PSA tables are stored in the default ODS tablespace.

3.4 Data Distribution of PSA Tables

Each transfer structure has a PSA table. Depending on the data load volume, the PSA table grows large in size. If a large size is expected, the PSA table should be distributed over several partitions.

When planning the data distribution of PSA tables, you have to take the following information into consideration:

- PSA tables with less than 10 GB can be stored on a single partition. You should create small PSA tables in the **same** tablespace.
- PSA tables larger than 10 GB should be distributed across multiple partitions.



We recommend that you create a separate database partition group for each large PSA table and a separate tablespace. This helps you to plan and to administer the space of these large PSA tables more individually and to create a separate buffer pool. For information about the creation of individual database partition groups and tablespaces, see [Creation of Large InfoCubes, ODS Objects, and PSA Tables \[Page 25\]](#).

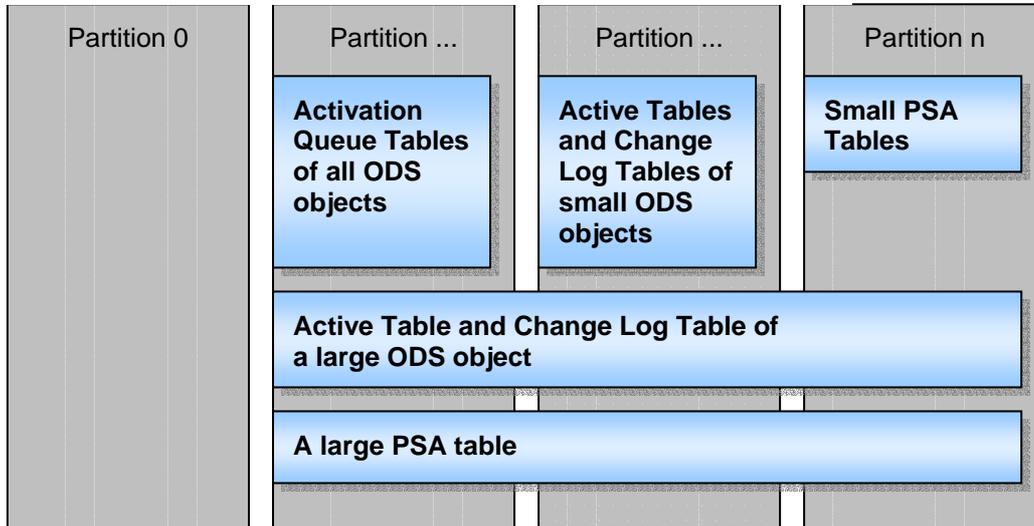
- Production ODS objects and PSA tables should be located in different tablespaces and not in the default ODS tablespace.



If you are using the BW default settings, ODS and PSA table are stored in the default ODS tablespace.

The following graphic shows an example of how you can distribute data of ODS objects and PSA tables:

Data Distribution of ODS Objects and PSA Tables



3.5 Partition Containing the Database Catalog

In a multi-partition installation, partition 0 contains the database catalog. When planning your database layout you should consider the following information:

- You always have to back up and restore partition 0, which contains the database catalog, **before** you can back up or restore all other partitions in parallel.
To minimize the backup or restore execution time of partition 0, you should reduce the amount of data stored on this partition. That is, you should avoid storing fact tables, ODS tables, or PSA tables on partition 0.
- In an environment with multiple database partition servers, you should reduce heavy workload on the database partition server containing the database catalog as described in the following example:



The database partition server A contains the database catalog and has the highest workload with a CPU utilization higher than 90 %. All other database partition servers have a workload with a CPU utilization of less than 40 %. To increase the total throughput of the SAP BW system, you can reduce the workload on the database partition server A by performing the following actions:

- You remove SAP BW work processes from database partition server A.
- You reduce partitions with fact, ODS, or PSA data on database partition server A and redistribute the data to partitions on other database partition servers.

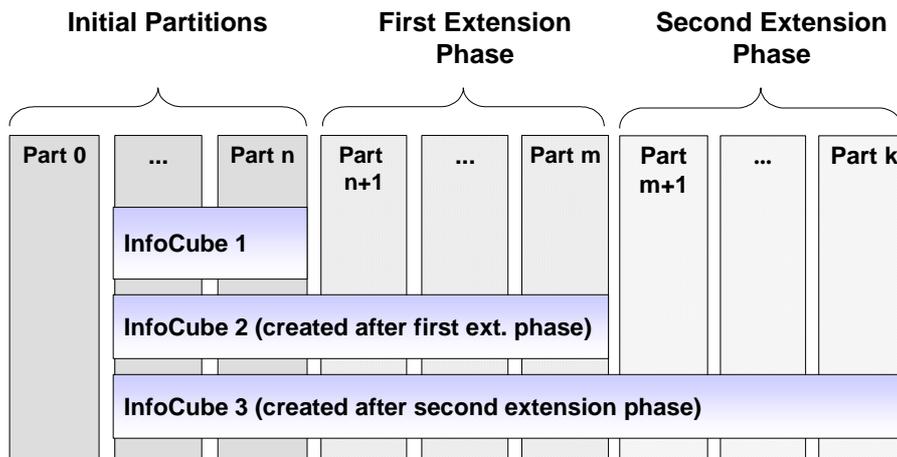
A heavy workload on database partition server A can increase the query response time if database processes on the other database partition servers are accessing tables (for example, dimension, master data or SAP system basis tables) that are located on database partition server A.

3.6 Scalable Partition Layout

Changing the number of partitions of a database partition group requires subsequent data redistribution. If you want to add new partitions to your BW system and create new InfoCubes, ODS objects, or PSA tables, you can avoid redistribution by creating new database partition groups and tablespaces for the new BW objects.

The following graphic shows how the database can grow with new InfoCubes and new partitions:

Database Growth with New InfoCubes



3 Database Layout Planning

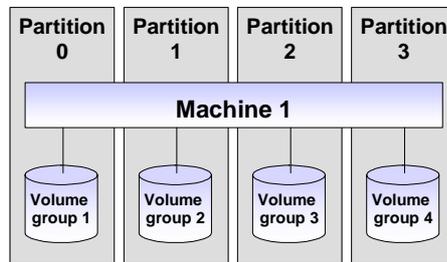
If you know that the maximum amount of data will be twice as much as in the first year of your business warehouse, you can start with twice the number of partitions you would need at the beginning (see the graphic below). Later on, you can add new database partition servers and move logical partitions to the new database partition servers. In this case, no redistribution is required.



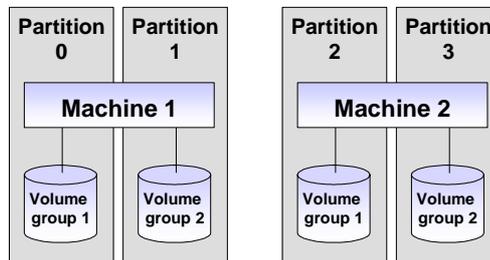
You have a system with an InfoCube with an initial size of 500 GB, but you plan to have a maximum data size of 1 Terabyte (TB). Initially, you have a four-way machine with four partitions. Later on, you add another machine and move two partitions to the new machine. On AIX, for example, this is easily possible because the file system layout has separate file systems for each partition, and the file systems for different partitions can be located in different volume groups.

Scaled Partition Layout

Initial Layout



Scaled Layout



4 Creation of Large Infocubes, ODS Objects, and PSA Tables

If you plan to create large InfoCubes, ODS objects or PSA tables, you should create separate database partition groups, separate buffer pools (if necessary), separate data tablespaces, separate index tablespaces and separate data classes for each new object in advance.

The advantages are as follows:

- You can maintain the physical disk layout separately for each object.
- You can add new partitions for single InfoCubes, ODS objects or PSA tables, and you can redistribute them one at a time.
- You can locate InfoCubes, ODS objects or PSA tables on different partitions.
- You can use separate buffer pools for individual InfoCube, ODS objects, or PSA tables.



You should consider adding new partitions to the database before you add a new InfoCube, ODS object, or PSA table. For more information, see [Adding New Partitions to the Database \[Page 36\]](#).

The following sections describe the database-related steps you have to perform when you create new InfoCube, ODS objects, or PSA tables on database partitions:

- [Creating Database Partition Groups \[Page 25\]](#)
- [Creating Buffer Pools \[Page 28\]](#)
- [Creating Tablespaces \[Page 29\]](#)
- [Creating Data Classes \[Page 33\]](#)

4.1 Creating Database Partition Groups

Use

Creating database partition groups is part of the overall process of creating large InfoCubes, ODS objects and PSA tables. A database partition group determines the partitions on which a tablespace is created.

Procedure

1. Identify the partitions where you create the large InfoCube, ODS object, or PSA table. For more information, see [Database Layout Planning \[Page 17\]](#).
2. Create the database partition group(s) for the InfoCube, ODS object, or PSA table as described in the following sections:
 - [Creating Database Partition Groups for InfoCubes \[Page 26\]](#)
 - [Creating Database Partition Groups for ODS Objects \[Page 27\]](#)
 - [Creating Database Partition Groups for PSA Tables \[Page 27\]](#)

4.1.1 Creating a Database Partition Group for InfoCubes

Use

You use the following procedure to create a database partition group for InfoCubes. The following two steps have to be performed:

1. You create a database partition group for fact tables.
2. You create a database partition group for aggregate tables.

You create a database partition group for aggregate tables to specify the single partition where the tablespace should reside **or** to specify a number of partitions for larger aggregates. You can create aggregates of different InfoCubes in the same tablespace on a single partition.

Procedure

Creating a Database Partition Group for Fact Tables

1. Log on to a database server as user db2<dbsid>.
2. Open a command prompt and enter the following SQL statement:

```
db2 create database partition group
NGRP_FACT_<SAPSID>#<INFOCUBE-SHORT-IDENTIFIER>
on dbpartitionnums(<partition_number>,
<second_partition_number>,..)
```

where <INFOCUBE-SHORT-IDENTIFIER> is the technical name of the InfoCube.

Creating a Database Partition Group for Aggregate Tables

To create a database partition group with **one partition** for aggregate tables, proceed as follows:

1. Log on to a database server as user db2<dbsid>.
2. Open a command prompt and enter the following SQL statement:

```
db2 create database partition group
NGRP_AGGR_<SAPSID>#<INFOCUBE-SHORT-IDENTIFIER>
on dbpartitionnum(<partition_number>)
```

where <INFOCUBE-SHORT-IDENTIFIER> is the technical name of the InfoCube.

4.1.2 Creating Database Partition Groups for ODS Objects

Use

As described in [Data Distribution of ODS Objects \[Page 20\]](#), you have to create a separate database partition group for the active table and the change log table of a large ODS object.

Procedure

1. Log on to a database server as user db2<dbsid>.
2. Open a command prompt and enter the following SQL statement:

```
db2 create database partition group
NGRP_ODS_<SAPSID>#AT_<ODS-SHORT-IDENTIFIER>
on dbpartitionnums (<partition_number>,
<second_partition_number>,..)
```

where <ODS-SHORT-IDENTIFIER> is the technical name of the ODS object.



If you want to store the activation queue tables of all ODS objects on a specific single partition, you create a database partition group to specify this partition by entering the following SQL statement:

```
db2 create database partition group
NGRP_ODS_<SAPSID>#ODS_QUEUES on dbpartitionnums
(<partition_number>)
```

4.1.3 Creating Database Partition Groups for PSA Tables

Use

PSA tables can grow very large in size. Therefore, partitioning of the PSA tablespaces considerably improves the performance of an SAP BW system when data is loaded from PSA.

Procedure

1. Log on to a database server as user db2<dbsid>.
2. Open a command prompt and enter the following SQL statement:

```
db2 create database partition group
NGRP_PSA_<SAPSID>#<PSA-SHORT-IDENTIFIER>
on dbpartitionnums (<partition_number>,
<second_partition_number>,..)
```

where <PSA-SHORT-IDENTIFIER> is the technical name of the PSA object.

4.2 Creating Buffer Pools

Use

You create buffer pools for individual InfoCubes, ODS and PSA tables to more precisely maintain the amount of main memory that is used to cache data pages for these objects.



Be aware that many buffer pools increase the effort of planning and maintenance as well as the risk that the main memory is fragmented. We therefore recommend that you first create buffer pools for the different types of data and share them with all data of the same type.

Procedure

The following procedure is an example of how to create a new buffer pool for the fact tablespace of a large InfoCube using an SQL statement or the DBA Cockpit:

SQL Statement

1. Log on to a database server as user db2<dbsid>.
2. Open a command prompt window and enter the following SQL statement:

```
DB2 CREATE BUFFERPOOL BP_FACT_<INFOCUBE-SHORT-IDENTIFIER>  
DATABASE PARTITION GROUP  
NGRP_FACT_<SAPSID>#<INFOCUBE-SHORT-IDENTIFIER> SIZE n  
PAGESIZE 16K
```

DBA Cockpit

For information about how to create buffer pools using the DBA Cockpit, see the SAP Online documentation and choose *SAP NetWeaver* → *Application Platform (SAP Web Application Server)* → *Databases* → *IBM DB2 Universal Database for UNIX and Windows* → *CCMS: SAP/DB2 UDB for UNIX and Windows* → *Configuration: Buffer Pools*.



You can create a buffer pool either on all partitions or on a subset of partitions that are defined by one or more database partition groups. You can only assign a tablespace to a buffer pool if the buffer pool exists on the same partition as the tablespace.



We recommend that you **explicitly** add the database partition group of a tablespace to the buffer pool to make sure that you can assign this tablespace to the buffer pool.

4.3 Creating Tablespaces

Use

You create new tablespaces as part of the overall process of [Creating Large Infocubes, ODS Objects and PSA Tables \[Page 25\]](#).

There are several advantages in separating large SAP BW tables of InfoCubes, ODS or PSA objects on different tablespaces:

- The size of tablespaces with 16 KB pages is limited to 256 GB per partition. If tables are located in the same tablespace, the size limit applies to the sum of these tables.
- Changing the tablespace disk layout is easier because you can perform the changes step by step with smaller data volumes.

You use and create managed-by-database (DMS) tablespaces for all types of data: dimension, fact, ODS, PSA and aggregate tables and indexes. We recommend that you either use file or raw device containers for these tablespaces.

Procedure

You create data and index tablespaces for InfoCubes, ODS objects, or PSA tables either using the following SQL statement or the DBA Cockpit.

SQL Statement

1. Log on to a database server as user db2<dbSID>.
2. Open a command prompt and enter the following SQL statement:


```
db2 create tablespace <SAPSID>#<TABLESPACE_NAME><T>
in database partition group <partition_group>
pagesize <pagesize> managed by database
using (file <container_1> <container_size>) on
dbpartitionnum(<partition_number_1>)
using (file <container_2> <container_size>) on
dbpartitionnum(<partition_number_2>)
...
using (file <container_n>) on
dbpartitionnum(<partition_number_n>)
extentsize <extentsize> prefetchsize <prefetchsize>
bufferpool <bufferpool>
```

The partition number should correspond to the 4-digit number in the container directory.



A rule of thumb is to create the index tablespace with half the size of the related data tablespace.

4 Creation of Large Infocubes, ODS Objects, and PSA Tables

3. To create new objects in the tablespaces, you have to grant the corresponding authorization to user `sap<sapsid>`. To do so, enter the following SQL statement:

```
db2 grant use of tablespace <SAPSID>#<TABLESPACE_NAME><T> to
user sap<sapsid>
```

Explanation of Variables

Variable	Description
<SAPSID>	3-character SAP system ID
<TABLESPACE_NAME>	Tablespace identifier that should correspond to the object identifier: <ul style="list-style-type: none"> • <INFOCUBE_SHORT_IDENTIFIER>, <ODS_SHORT_IDENTIFIER> • <PSA_SHORT_NAME_IDENTIFIER>
<T>	Type of tablespace: <ul style="list-style-type: none"> • 'D' for data • 'I' for index
<partition_group>	Name of the database partition group as created in Creating Database Partition Groups [Page 25]
<pagesize>	We recommend that you have 16 KB for fact, aggregate, ODS and PSA tablespaces (data and index).
<container_x>	File name of the DMS container: <ul style="list-style-type: none"> • UNIX: '/db2/<SAPSID>/sapdata<mm>/NODE<nnnn>/<SAPSID>#<TABLESPACE_NAME><T>.container<ccc>' • Windows: '<drive>:\db2\<SAPSID>\sapdata<mm>\NODE<nnnn>\<SAPSID>#<TABLESPACE_NAME><T>.container<ccc>' where <drive> is the drive letter of the logical drive.

<mm>	<p>Number of the <code>sapdata</code> directory. The default values are as follows:</p> <ul style="list-style-type: none"> • 7 Dimension data and index tablespace containers • 8 Fact data and index tablespace containers • 9 ODS and PSA data and index tablespace containers <p> UNIX only: If you create new file systems Use numbers higher than 9.</p>
<nnnn>	4-digit partition number, for example, 0001 for partition 1.
<ccc>	3-digit container number of this tablespace container on the local partition
<container_size>	Size of the container
<partition_number_x>	Number of the partition where the container is located. It should correspond to <nnnn> .
<extentsize>	Recommended value is 16 for all types of tablespaces.
<prefetchsize>	Recommended value is 16 for all types of tablespaces.
<bufferpool>	Name of the buffer pool this tablespace should be assigned to



We recommend that you use the conventions for tablespace names, container directories, and container file names as described in the table above.

DBA Cockpit

For information about how to create tablespaces using the DBA Cockpit, see the SAP Online documentation and choose *SAP NetWeaver → Application Platform (SAP Web Application Server) → Databases → IBM DB2 Universal Database for UNIX and Windows → CCMS: SAP/DB2 UDB for UNIX and Windows → Space: Tablespaces.*

Example

In the SAP BW system `BWP` in the database partition group `NGRP_FACT_BWP#IC01`, you want to create a new DMS fact tablespace for the fact tables of the InfoCube `IC01`. You then want to assign this tablespaces to buffer pool `BP_BW_16K`. To do so, proceed as follows:

1. Log on to a database server as user `db2<dbssid>`.
2. Open a command prompt and enter the following SQL statement:

```
db2 CREATE TABLESPACE BWP#IC01D IN DATABASE PARTITION GROUP
NGRP_FACT_BWP#IC01 PAGESIZE 16K MANAGED BY DATABASE

USING (FILE '/db2/BWP/sapdata10/NODE0000/BWP#IC01D.container000'
25G) ON DBPARTITIONNUM (0)

USING (FILE '/db2/BWP/sapdata10/NODE0000/BWP#IC01D.container001'
25G) ON DBPARTITIONNUM (0)

USING (FILE '/db2/BWP/sapdata10/NODE0001/BWP#IC01D.container000'
25G) ON DBPARTITIONNUM (1)

USING (FILE '/db2/BWP/sapdata10/NODE0001/BWP#IC01D.container001'
25G) ON DBPARTITIONNUM (1)

EXTENTSIZE 16 PREFETCHSIZE 16 BUFFERPOOL BP_BW_16K
```

The fact tablespace is stored in two containers on two partitions. On each partition, the containers are located in the `sapdata10` and `sapdata11` subdirectories. The size of each container is 25 GB. The short name for the tablespace is `IC01`.

Constraints

If you are **not** using a storage system, you can avoid I/O bottlenecks by applying the disk striping function of DB2. To do so, create a tablespace with several containers of the same size, each one located on a separate disk. If the number of disks is limited, index and data tablespace containers can reside in pairs on the same disk as illustrated in the following table:

	Disk 1	Disk 2	Disk n
Data Tablespace 1	container 1	container 2	container n
Index Tablespace 1	container n	container 1	container 2
Data Tablespace 2	container 1	container 2	container n
Index Tablespace 2	container n	container 1	container 2

To avoid that data and corresponding indexes are read from the same disk, the data tablespace containers are located on the disks in a different order than the index tablespace containers.

4.4 Creating Data Classes

Use

A data class is the counterpart of a tablespace in the SAP system. It associates a data tablespace with an index tablespace and defines a logical name for this combination. This logical name is referenced in the ABAP Dictionary and in the BW metadata.

To make the previously created tablespaces known to the SAP BW system, you create new data classes by defining a data class for the fact tables, aggregates, ODS and PSA tablespaces according to the default data classes DFACT and DODS.

You assign these data classes in the BW Administration Workbench when you create a new InfoCube, ODS object, or PSA table as described in the following sections. The assigned data classes determine the tablespaces and the corresponding data partition groups.

Procedure

To create, that is add a new data class, proceed as follows:

1. In your SAP BW system, call transaction DB6COCKPIT.
2. In the navigation frame, choose *Configuration* → *Data Classes*.
The *Configuration: Data Classes* screen appears
3. To add a new data class, choose *Add*.
The *Add Data Class* dialog box appears.
4. Enter the name and description of the new data class. The data class name should be in compliance with the information in **SAP Note 46272**.

You should use the following prefixes for the data classes:

- Fact table: ZF*
- Dimension table: ZD*
- Aggregate table: ZA*
- ODS Active table and change log table: ZO*
- ODS Activation queue table: ZQ*
- PSA table: ZP*



Keep in mind that not defining data classes according to the naming conventions has an impact on future upgrades of your system. Such entries will not be recognized as customer entries and will be lost during the upgrade.

5. From the list box, select a data and index tablespace for the new data class.
6. To confirm your entries, choose *Add*.

See also:

- [Assigning InfoCubes and their Aggregates to Data Classes \[Page 34\]](#)
- [Assigning ODS Objects to Data Classes \[Page 35\]](#)
- [Assigning PSA Tables to Data Classes \[Page 35\]](#)

4.4.1 Assigning InfoCubes and Their Aggregates to Data Classes

Use

The following describes the two steps that you have to perform to assign InfoCubes and their aggregates to data classes.

By default, the fact tables of an InfoCube and its aggregates are created in the tablespace that is defined in the data class DFACT. By default, the dimension tables are created in the tablespace that is defined in the data class DDIM. For each InfoCube that should be stored in a separate tablespace, you need to overwrite the default data class with the new data class.



You can assign new data classes to fact tables of an InfoCube as long as the InfoCube does not contain any data.

Procedure

Changing the Data Class of Infocubes

1. In your SAP BW system start the *Administrator Workbench*.
2. Create a new InfoCube or choose the required InfoCube and choose *Change* from the context menu.
The *Edit InfoCube* screen appears.
3. Choose *Extras* → *Maintain DB-storage parameters*.
The *Maintain Storage Parameters* dialog box appears.
4. In the *InfoCube: Logical Storage Parameter* area, change the data class for fact tables.
5. If you created separate tablespaces for the InfoCube or Aggregate dimension tables, change the value in the *Data class* field to the new data class for dimension tables.
6. Once you have completed all entries, choose *Activate*.
When you activate the InfoCube, the tables are created in the tablespaces when you activate the InfoCube.

Changing the Data Class of Aggregates

By default, the aggregate fact tables of an InfoCube are created in the same data class as the fact tables of the InfoCube. To determine the default fact table data class for all subsequently created aggregates of an InfoCube, proceed as follows:

1. In your SAP BW system start the *Administrator Workbench* and choose the required InfoCube and choose *Maintain Aggregates....* from the context menu.
The *Maintenance for Aggregate* screen appears.
2. Choose *Extras* → *Change Data Class for Aggregates*.
The *Choose Data Class for Aggregates for InfoCube <InfoCube_name>* appears.
3. Choose the new data class for aggregate tables and confirm your entries.
4. Once you have completed all entries, choose *Activate*.



If you want to change the data class of already filled aggregates, you have to deactivate the aggregates first. Then re-activate and fill them again.

4.4.2 Assigning ODS Objects to Data Classes

Use

By default, all tables of an ODS object are created in the tablespaces that were defined in the data class DODS. You can change the default data class of the active table and the data class of the activation queue table as long as the ODS object does not contain any data.

Procedure

1. In your SAP BW system, start the *Administrator Workbench*.
2. Create a new ODS object or choose the required ODS object and choose *Change* from the context menu.
The *Edit ODS Object* screen appears.
3. Choose *Extras* → *Maintain DB-storage parameters*.
The *Maintain Storage Parameters* dialog box appears.
4. In the *Log. Storage Parameter for ODS Object (Active Records)* area, choose a new data class.
5. In the *Log Storage Parameter for ODS Object (Activation Queue)* area, choose a new data class.
6. Confirm your entries.
7. Once you have completed all entries, choose *Activate*.

4.4.3 Assigning PSA Tables to Data Classes

Use

By default, PSA tables are created in the data class DODS. You can change the data class of a PSA table if the corresponding *DataSource* has been assigned to an *InfoSource*. Once the transfer structure of the *InfoSource* has been activated, you **cannot** change the data class of the PSA table.



A PSA table is created when the *Transfer Structure* of an *InfoSource* is activated.

Procedure

1. In your SAP BW system start the *Administrator Workbench*.
2. Assign a *DataSource* to an *InfoSource* object.
The *InfoSource <name> Change* screen appears.
3. Choose *Extras* → *Maintain DB-storage parameters*.
The *Maintain Storage Parameters for PSA Table* dialog box appears.
4. Choose the new data class for PSA tables and confirm your entries.
5. To activate the transfer structure, choose *Activate*.

5 Adding of New Partitions to the Database

Purpose

The following section describes the steps that are necessary if you want to add new partitions to the database.

Process Flow



Before you start adding partitions, we strongly recommend that you perform a complete backup of the database.

1. You create new partitions and perform all necessary steps as described in [Preparing New Partitions \[Page 37\]](#).
2. If you want to add new, large InfoCubes, ODS objects, or PSA tables to the system and if you want to create new database partition groups for these objects, proceed with [Creating Large InfoCubes, ODS Objects, and PSA Tables \[Page 25\]](#).
3. If you want to redistribute InfoCubes, ODS objects, or PSA tables to the new partitions, you can either perform this step by using the table conversion report DB6CONV or using the `DB2 REDISTRIBUTE DATABASE PARTITION GROUP` command:

- **Report DB6CONV**

You use report DB6CONV for the conversion of single tables. It is well suited for single, large InfoCubes, ODS, and PSA objects in separate tablespaces. For each table, a background job in the SAP system is scheduled. For a large number of tables, the manual effort for scheduling all conversions can be high.

For BW tables, report DB6CONV supports the conversion option *not logged initially*. The conversion is not written to the log files, that is, this conversion method **saves execution time** and **log space**. However, you must perform a complete backup after the conversion has finished.

For detailed information about report DB6CONV, see **SAP Note 362325**. If you want to use this report, see [Redistributing the Data with DB6CONV \[Page 42\]](#).

- **DB2 REDISTRIBUTE**

You use the `DB2 REDISTRIBUTE` command to redistribute all tables in tablespaces in the specified database partition group. You should use the `DB2 REDISTRIBUTE` command for tablespaces with a large number of tables. `DB2 REDISTRIBUTE` requires free log space of at least the amount of the largest distributed table since each table is redistributed in one unit of work. Therefore, you might have to increase the active log space.

If you want to use this command, see [Redistributing the Data with DB2 REDISTRIBUTE \[Page 46\]](#).



When partitions are added to an existing database partition group and tables exist in this group, the new partitions are not added automatically to the partition map of the group. In this case, you **must** use the **DB2 REDISTRIBUTE** command because only this command is able to update partition maps. You **cannot** use report DB6CONV in this case. For the redistribution with DB6CONV, you have to create a new database partition group with a partition map that includes all new partitions and a new tablespace in this group. The partitioned tables are then moved to the new tablespaces created in the new database partition group.



No matter which option you use, be aware that tables **cannot** be accessed during the whole redistribution process.

5.1 Preparing New Partitions

To prepare new partitions, proceed as follows:

1. [Create additional partitions \[Page 37\]](#).
2. [Update the database configuration for each new partition \[Page 39\]](#).
3. [Add containers to the system temporary tablespaces on each new partition \[Page 41\]](#).

5.1.1 Creating Additional Partitions

Use

You use one of the following procedures to create additional partitions in a multi-partition environment, that is you:

- Add new database partition servers to a multi-partition installation
- Create additional logical partitions on already existing database partition servers

Procedure

Adding New Database Partition Servers

To add new database partition servers to a multi-partition installation, you have to install a database instance on each new computer. For more information, see the SAP documentation *Installation Guide – SAP Web Application Server ABAP 6.40 on <Operating system platform>: IBM DB2 Universal Database for UNIX and Windows*, section *Installing SAP BW on Multiple Database Partition Servers*. This documentation is included in the installation package or available on SAP Service Marketplace at

service.sap.com/instguidesNW04 → *Installation*.

Creating Additional Logical Partitions

1. Log on to the instance-owning database partition server as user `db2<dbsid>`.
2. **UNIX only:**
Verify that all participating database partition servers have an entry in the `.rhosts` file that is located in the database instance directory `/db2/db2<dbsid>`.
3. Open a DB2 command window and enter the following command, one for each partition to be added to the database manager instance:

- **UNIX:**

```
db2start dbpartitionnum <partition_number> add
dbpartitionnum hostname <computer_name> port <logical_port>
without tablespaces
```

- **Windows:**

```
db2start dbpartitionnum <partition_number> add
dbpartitionnum hostname <computer_name> port <logical_port>
computer <computer_name> user <user_name> password
<password> without tablespaces
```



If the SQL error SQL6030N: START or STOP DATABASE MANAGER failed. Reason code 14 occurs, check the port values in the system's services file. Open the file `<services file>` and search for `DB2<DBSID>`. Two entries `DB2_DB2<DBSID>` and `DB2_DB2<DBSID>_END` define the port value interval for all partitions in the multi-partition installation. If the number of partitions exceeds this interval increase the value of `DB2_DB2<DBSID>_END` by the number of new partitions.

4. Stop and restart the database manager (`db2stop; db2start`).

Explanation of Variables

Variable	Description
<code><dbsid></code>	Database name
<code><partition_number></code>	Partition number of the new partition (If four partitions 0, 1, 2, 3 exist, the number is 4)
<code><computer_name></code>	Name of the database partition server on which the new partition is to be created
<code><logical_port></code>	Partition number on the particular database partition server. If this is the first partition on the database partition server, the logical port is 0. You can also find the number in the <code>db2nodes.cfg</code> file.
<code><user_name></code>	Name of the default database administration user <code>db2<database_name></code> If the user is created in a Windows Domain, you also have to specify the domain: <code><domain>\<user_name></code>

<password>	User password
<services_file>	UNIX: /etc/services Windows: %SystemRoot%\system32\drivers\etc\services

5.1.2 Updating the Database Configuration for Each New Partition

Use

Once you have created additional partitions, you have to update the database configuration.

In a multi-partition installation, each partition has its own database configuration. The database configuration parameters of a new partition are set to DB2 default values.



Update the parameters according to the **SAP Note 584952** (DB2 UDB Version 8.1) and your experiences. The optimal configuration depends on the database layout and on the data distribution and may result in different database configurations for the participating partitions. A general recommendation in addition to the mentioned SAP Note **cannot** be given.

Prerequisites

Before you update the database configuration parameters, you can check the database configuration of a partition using one of the following commands:

- You can check the database configuration by entering the following CLP command:

```
db2_all "<<+<partition_number>< db2 get database configuration
for <database_name>"
```
- You can also check the database configuration by setting the environment variable DB2NODE to the desired partition number. To do so, enter the following commands:

```
set DB2NODE=<partition_number>
db2 get database configuration for <database_name>
```



After setting the DB2NODE environment variable, all DB2 commands are executed on the specified partition.

Procedure

To update the database configuration parameters, we recommend that you write an SQL script that you execute on each new partition:

1. Log on to the instance-owning database partition server as user `db2<dbssid>`.
2. To create the SQL script in file `<dbconfig_file>`, add the following line for each parameter:

```
update database configuration for <database_name> using  
<parameter_name> <value>;
```



It is important that you make sure to set the parameter `NEWLOGPATH` to `<log_path>`.

3. To update the database configuration, enter the following command for each new partition:

```
db2_all "<+<partition_number>< db2 -tf <dbconfig_file>"
```

4. To stop and restart the database manager, use the following commands:

```
db2stop
```

```
db2start.
```

5. To connect to the database, enter the following command:

```
db2 connect to <database_name>
```

This may take some time since the log files for all new partitions are created in the specified log directory.

Explanation of Variables

Variable	Description
<code><dbconfig_file></code>	Name of the file created by the user to update the database configuration parameters
<code><log_path></code>	SAP standard log path: <ul style="list-style-type: none">• UNIX: "/db2/<SAPSID>/log_dir/"• Windows: "<drive>:\db2\<SAPSID>\log_dir\"

5.1.3 Adding Containers to Temporary Tablespaces on Each New Partition

1. Log on to a database server as user db2<dbsid>.
2. To add containers for each new partition, enter the following SQL statements:

```
db2 alter tablespace PSAPTEMP add ('<temp_containers>') on
dbpartitionnum(<partition_number>)
```

```
db2 alter tablespace PSAPTEMP16 add ('<templ6_containers>') on
dbpartitionnum(<partition number>)
```

The 4-digit-number in the container directory should correspond to the partition number.

Explanation of Variables

Variable	Description
<temp_containers>	<p>SAP standard path for tablespace containers of the temporary system tablespace with a page size of 4 KB.</p> <ul style="list-style-type: none"> • UNIX: "/db2/<SAPSID>/saptemp1/NODE<nnnn>/temp4" • Windows: "<drive_letter>:\db2\<SAPSID>\saptemp1\NODE<nnn>\temp4"
<templ6_containers>	<p>SAP standard path for tablespace containers of the temporary system tablespace with a page size of 16 KB.</p> <ul style="list-style-type: none"> • UNIX: "/db2/<SAPSID>/saptemp1/NODE<nnnn>/templ6" • Windows: "<drive_letter>:\db2\<SAPSID>\saptemp1\NODE<nnn>\templ6"
<nnnn>	4-digit partition number, for example, 0001 for partition 1

5.2 Redistributing the Data with DB6CONV

Use

As described in [Adding New Partitions to the Database \[Page 36\]](#), you use report DB6CONV to convert tables and redistribute data over multiple partitions.

Procedure

To redistribute the data with DB6CONV, you proceed as follows:

1. [Create new database partition groups \[Page 42\]](#).
2. [Create new tablespaces in the new database partition groups \[Page 43\]](#).
3. [Check the partitioning keys of all partitioned tables \[Page 43\]](#).
4. [Redistribute all partitioned tables \[Page 44\]](#).

5.2.1 Creating New Database Partition Groups

Use

When redistributing the data with DB6CONV, you **cannot** add new partitions to the partition maps of existing database partition groups. You have to create new database partition groups that include the new partitions in the partition map and then add the new database partition group to the corresponding buffer pool.

To redistribute all partitioned tables, you create a new group for each database partition group. Then move all tables from the original group to the new group using report DB6CONV, and drop the original groups.

Procedure

1. Create new database partition groups as described in section [Creating Database Partition Groups \[Page 25\]](#).
2. Add the new database partition groups to the corresponding buffer pools as described in the following example:



To add the database partition group <partition_group> to the default buffer pool for fact tables BP_BW_16K , log on to a database server as user db2<dbsid> and enter the following SQL statement:
db2 alter bufferpool BP_BW_16K add database partition group <partition_group>
where <partition_group> is the name of the database partition group.

5.2.2 Creating New Tablespaces in the New Database Partition Group

Use

DB6CONV moves the tables from the current tablespaces to the target tablespaces that were created in the new database partition groups. You have to create a new tablespace for each original tablespace. Then you move all tables from the original tablespace to the new tablespace and drop the original tablespace.

Procedure

1. Create new tablespaces as described in section [Creating Tablespaces \[Page 29\]](#).



Always create the data tablespace and the corresponding index tablespace.

If you replace original tablespaces with new tablespaces, change the corresponding data classes in the SAP system. To avoid inconsistency with the metadata of BW objects, you should not drop or rename data classes. Hence, in the existing data classes, replace the original tablespaces with the new tablespaces rather than creating new data classes and dropping the old data classes. This ensures later that new BW objects are created in the correct tablespaces.

2. Change data classes as described in [Creating Data Classes \[Page 33\]](#).

5.2.3 Checking the Partitioning Keys of All Partitioned Tables

Use

You **only** use the following procedure if your tables are not evenly distributed in a multi-partition SAP BW system.

Adding new partitions neither invalidates partitioning keys nor it requires changing the partitioning keys. If you migrate from a database system that does not support hash partitioning, incorrect or missing partitioning keys can occur after a heterogeneous system migration.



We strongly recommend that you perform these checks after a heterogeneous system migration. If you are not sure about the status of partitioning keys or if you have a single-partition system, you should perform these checks as well.

To find out whether all partitioned tables are distributed evenly across the partitions defined in the database partition groups, you can check and repair the partitioning key of each partitioned table using the following function modules:

- RSDU_CHKREP_PKEY_ALL_CUBES_DB6
- RSDU_CHKREP_PKEY_ALL_ODS_DB6
- RSDU_CHKREP_PKEY_ALL_PSA_DB6

For more information about these function modules, see **SAP Note 648432**.

Procedure

1. Log on to the SAP BW system.
2. Call transaction SE37.
The *Function Builder: Initial Screen* appears.
3. Enter the function module name `RSDU_CHKREP_PKEY_ALL_CUBES_DB6` and choose *Single Test*.
The *Test Function Module:Initial Screen* appears.
4. For parameters `I_PROC_AGGREGATES` and `I_PREP_REPAIR`, enter **X**
5. Choose *Execute*.
The *Test Function Module: Result Screen* appears.
6. Go back to the *Function Builder: Initial Screen* and repeat steps 2 to six for the modules `RSDU_CHKREP_PKEY_ALL_ODS_DB6` and `RSDU_CHKREP_PKEY_ALL_PSA_DB6`.

There are two scenarios that you have to take into account:

- If the source tablespace is on a single partition, missing partitioning keys are created and wrong partition keys are repaired immediately.
- If the source tablespace is distributed on two or more partitions, partitioning keys **cannot** be repaired without copying the table content. In this case, the check function modules create entries for the distributed tables with incorrect partitioning keys in table `DB6CONVT`. You then have to run report `DB6CONV` to convert the tables.

By default, the source and target tablespaces for the repair conversions are the same. Therefore, if you have to repair missing or incorrect partitioning keys of tables and you want to redistribute these tables to a new database partition group, you have to run `DB6CONV` twice. The first run corrects the partitioning keys and the second run redistributes and moves the tables to the new tablespaces.

You can perform the actions *repair* and *redistribute* in a single execution of `DB6CONV`, which is especially helpful for very large tables whose conversion may take a long time. For a single execution, call report `DB6CONV` and edit the `DB6CONVT` entry for all tables that you want to repair and redistribute in a single step. Then change the fields *Target Data Tablespace* and *Target Index Tablespace* to the names of the new tablespaces, which you previously created as described in [Creating New Tablespaces in the New Database Partition Group \[Page 43\]](#).

5.2.4 Redistributing All Partitioned Tables

Use

To schedule tables for conversion with `DB6CONV`, you have to make entries in table `DB6CONVT`. To create entries for all partitioned tables in a source tablespace, you can use function module `RSDU_REPARTITION_PREPARE_DB6`. For more information, see **SAP Note 648432**.



Function module `RSDU_REPARTITION_PREPARE_DB6` schedules all tables with the option ***not logged initially***.

Procedure

1. Log on to the SAP BW system.
2. Call transaction SE37.
The Function Builder: Initial Screen appears.
3. Enter the function module name `RSDU_REPARTITION_PREPARE_DB6` and choose *Single Test*.
The Test Function Module :Initial Screen appears.
4. Enter the name of the source tablespace in parameter field `I_TABLESPACE` and the name of the target tablespace in parameter field `I_TABLESPACE_NEW`.
5. Choose *Execute*.



A data class must exist that contains the target tablespace and its corresponding index tablespace. You can edit the data classes to contain the new tablespaces **before** you call this function module. Index tablespaces do not need to be specified as source tablespaces since they do not contain any tables.

6. Call transaction SE38.
7. Enter the report name `DB6CONV` and choose *Execute*.
8. To schedule one or more DB6CONV jobs, select one or more tables with status *NEW*.
9. To start the redistribution, choose *Convert NEW*.



Since the redistribution of the tables is **not** logged, you **must** perform a complete backup after all tables have been redistributed. You also have to schedule a `runstats_all` job afterwards to update the database statistics for all redistributed tables.

5.3 Redistributing the Data with DB2 REDISTRIBUTE

Use

The process of redistributing data using the DB2 REDISTRIBUTE command is similar to redistributing data using report DB6CONV. However, instead of creating new database partition groups in a first step and creating new tablespaces in a second step, the database partition groups and tablespaces are altered.

Procedure

To redistribute the data with DB2 REDISTRIBUTE, you have to proceed as follows:

1. [Add partitions to database partition groups \[Page 46\]](#).
2. [Add tablespace containers on each new partition \[Page 47\]](#).
3. [Check the partitioning keys of all partitioned tables \[Page 47\]](#).
4. [Redistribute the database partition groups \[Page 48\]](#).

5.3.1 Adding Partitions to Database Partitions Groups

1. Log on to a database partition server as user db2<dbssid>.
2. To add one ore more partitions to a database partition group, open a command prompt and enter the following SQL statement:

```
db2 alter database partition group <partition_group>
add dbpartitionnums (<partition_number>[,
<second_partition_number>, ...]) without tablespaces
```



Do **not** omit the option `without tablespaces`. Otherwise, DB2 default containers, which differ from the SAP standard, are created on the new partitions for each tablespace that is defined in the database partition group. Tablespace containers in the new partitions are created explicitly in the next step.

5.3.2 Creating Tablespace Containers on Each New Partition

Use

On each partition that is added to a database partition group, you have to create tablespace containers for all tablespaces that belong to the database partition group.

Procedure

1. Log on to a database server as user db2<dbsid>.
2. To create the required tablespace containers, enter the following SQL statement:

```
db2 alter tablespace <table_space> add (file '<container>'
<size>) on dbpartitionnum (<partition_number>)
```

Explanation of Variables

Variable	Description
<table_space>	Tablespace name
<container>	Path or file name of tablespace container
<size>	Size of the new container

5.3.3 Checking The Partitioning Keys of all Partitioned Tables

Use

You **only** use the following procedure if your tables are not evenly distributed in a multi-partition SAP BW system.

Adding new partitions does neither invalidate partitioning keys nor requires changing the partitioning keys. If you migrate from a database system that does not support hash partitioning, incorrect or missing partitioning keys can occur after a heterogeneous system migration,



We strongly recommend that you perform these checks after a heterogeneous system copy. If you are not sure about the status of partitioning keys or if you have a single-partition system you should perform these checks as well.

To find out whether all partitioned tables are distributed evenly across the partitions defined in the database partition groups, you can check and repair the partitioning key of each partitioned table with the following function modules:

- RSDU_CHKREP_PKEY_ALL_CUBES_DB6
- RSDU_CHKREP_PKEY_ALL_ODS_DB6
- RSDU_CHKREP_PKEY_ALL_PSA_DB6

For more information about these function modules, see **SAP Note 648432**.

Procedure

1. Log on to the SAP BW system.
2. Call transaction SE37.
The Function Builder: Initial Screen appears.
3. Choose function module `RSDU_CHKREP_PKEY_ALL_CUBES_DB6` and *Single Test*.
The Test Function Module: Initial Screen appears.
4. For both parameters `I_PROC_AGGREGATES` and `I_PREP_REPAIR` enter **x**
5. Choose *Execute*.
6. Repeat this procedure with the modules `RSDU_CHKREP_PKEY_ALL_ODS_DB6` and `RSDU_CHKREP_PKEY_ALL_PSA_DB6`.

5.3.4 Redistributing Database Partition Groups

Use

You use the following procedure to redistribute all tables in tablespaces belonging to a database partition group.

Procedure

1. Log on to a database partition server as user `db2<dbssid>`.
2. To redistribute database partition groups, enter the following CLP command:

```
db2 redistribute database partition group <partition_group>  
uniform
```



The following SQL error can occur although you have checked and repaired tables (as described in [Checking The Partitioning Keys of all Partitioned Tables \[Page 47\]](#)):

```
SQL6047N: Nodegroup cannot be redistributed because table  
name does not have a partitioning key
```

Use report DB6CONV to move all tables that do not have a partitioning key to a non-distributed tablespace for example, the tablespace for dimension tables.

After successfully redistributing database partition groups, perform a reorganization of all redistributed tables to free empty data pages on the source partitions. For reorganization, use the action *REORG of Tables in Tablespace(s)* in the DBA Planning Calendar. It is also necessary to schedule a *runstats_all* job afterwards. For more information see the SAP Library and choose *SAP NetWeaver* → *Application Platform (SAP Web Application Server)* → *Databases* → *IBM DB2 Universal Database for UNIX and Windows* → *CCMS: SAP/DB2 UDB for UNIX and Windows*.

6 Best Practices

6.1 Data Clustering of InfoCube Fact Tables

Concern

The queries on the base fact tables of InfoCubes have a long execution time because the qualifying rows are spread across many data pages in the database.

Goal

To reduce the I/O effort during query processing and to improve the query performance, the respective rows should be clustered in the physical data pages.

Solution

DB2 UDB for UNIX and Windows can sort rows in physical data pages in the order of an index that is also known as the clustering index. The clustering index is created automatically for fact tables and is a compound index containing all dimension key columns. The order of the clustering index is determined by the InfoCube layout, respectively by the order of the InfoCube dimensions.

The InfoCube layout is defined in the SAP BW Administrator Workbench during the InfoCube creation. The InfoCube dimensions are created during the InfoCube creation and SAP BW characteristics are assigned to InfoCube dimensions.

The clustering index has the following order according to the defined dimensions:

1. Time dimension
2. Dimension 1
3. Dimension 2
4. Dimension 3
5. Dimension <n>

Thus, the assignment of the characteristics to InfoCube dimensions determines the data clustering of fact tables in the database.



We recommend that you define the dimension with the highest expected cardinality as the first dimension, the dimension with the second highest cardinality as the second dimension and so on. This order guarantees a good clustering of the fact data.

Data clustering groups records with the same key value. Data with the same key is spread over less data pages and as a result, less data pages must be read during a BW query (for example, with given restrictions on a characteristic that is assigned to the first dimension). Thus, data clustering can improve the performance of a BW query.

Example

The following describes how you can improve the performance of an SAP BW query and thus of an SAP BW system by applying the correct data clustering:

The given characteristics are as follows:

- 50000 Customers
- 10000 Materials
- 10 Sales Organizations
- 50 Departments.

A suitable InfoCube dimension assignment would be as follows:

- Dimension 1: Customer
- Dimension 2: Material
- Dimension 3: Department
- Dimension 4: Sales Organizations

The query selects all materials that a specific customer acquired last year at a given time. On average, 2000 rows per customer are selected. Depending on how you defined the customer characteristic, the following is going to happen:

- If the *Customer* characteristic is defined as the last dimension, 2000 rows have to be selected for the query from all over the fact table.
This results in bad query performance, because up to 2000 data pages must be read.
- If the *Customer* characteristic is defined as the first dimension, 2000 rows have to be selected from only a few data pages of the fact table.
The query performance will be much better because the same amount of rows has to be selected from a few data pages only.

6.2 Controlling Database Log Space Consumption

Concern

A SAP BW function stops with the error message that the database log space is exceeded.

Goal

You want to reduce the amount of database log space that is consumed by a BW function.

Solution

Certain operations in an SAP BW system require a considerable amount of log space in the database. You can control the log space consumption of these operations by setting the corresponding BW parameters, which belong to table RSADMINC or ROIDOCPRMS, as described in the following list:

- Function *Aggregate Build / Change Run*:

You have to set parameter `BLOCKSIZE` in table RSADMINC. In your SAP BW system call transaction SPRO and choose *SAP Reference IMG → Business Information Warehouse → General BW settings → Parameters for aggregates*.

If not defined (value 0), the default value of parameter `BLOCKSIZE` is 100,000,000. The unit of measurement is "records".



If you are running out of log space during this operation, we recommend that you lower this value, for example to 1,000,000. Very small values can have a negative impact on performance.

- Function *Loading InfoPackage from file into PSA or InfoCube*:

You have to set parameter `IDOCPACKSIZE` in table RSADMINC. In your SAP BW system, call transaction SPRO and choose *SAP Reference IMG → Business Information Warehouse → Links to Other Systems → Maintain Control Parameters for the data transfer → Package Size*.

Parameter `IDOCPACKSIZE` determines how many records are inserted in one transaction. A reasonable value for `IDOCPACKSIZE` is 50,000. Very small values can have a negative impact on performance.

- Function *Loading InfoPackage from R/3 Source System into PSA and/or InfoCubes*:

You have to set parameter `MAXSIZE` in table ROIDOCPRMS in the source system. For more information about how to set this parameter, see **SAP Note 417307**.

Parameter `MAXSIZE` determines the size of a data package. The unit of measurement is "kilobytes".



If you are using parallel load processes and load data into several data targets concurrently, log space consumption will increase.

- Function *InfoPackage Deletion*:

You have to set parameter `DB6_DELETE_PIECEWISE` in table `RSADMIN`. The deletion of InfoPackages also happens during compression. For more information about how to set parameter `DB6_DELETE_PIECEWISE`, see **SAP Note 597053**.



If the parameter is set to *YES*, data is deleted in blocks of 100,000 records.

6.3 Dropping Indexes Before Deletion and Compression of InfoPackages

Concern

The deletion of InfoPackages from InfoCubes requires a long execution time. The same applies during compression when InfoPackages are deleted.

Goal

You want to speed up the deletion of InfoPackages and thus improve the performance of your SAP BW system.

Solution

The performance of `INSERT` and `DELETE` SQL statements is affected by the maintenance of indexes that exist on the tables. This specifically applies to data manipulation operations that affect a large number of rows on tables with several indexes.

Fact tables of InfoCubes have one index defined on each dimension and a compound index over all dimensions.

Maintaining the indexes decreases the overall performance of loading and deleting large InfoPackages in such a way that it is faster to drop the indexes before the operation and to rebuild the indexes afterwards.

SAP BW provides functions to drop and to recreate the secondary indexes (the indexes on a single dimension of both fact tables of an InfoCube).



The compound index over all dimensions, also known as **P-index**, is **never** dropped. It ensures the uniqueness on the compressed E fact table and the clustering of both fact tables.

The dropping and recreation of secondary indexes can be automatically applied when InfoPackages are loaded by switching on the flag *Delete InfoCube indexes before each data load and then refresh*.

To do so, proceed as follows:

1. In your SAP BW system start the *Administrator Workbench*.
2. Select the required InfoCube and choose *Manage* from the context menu.

The *Manage Data Targets* screen appears.

3. On the *Performance* tab, choose *Create Indexes (batch)*.
The *Create index for InfoCube <infocube_name>* dialog box appears.
4. Check that *Delete InfoCube indexes* is selected **before** each data load.
5. Choose *Refresh* and confirm your entries.

You can improve the overall performance of InfoPackage deletion or compression by dropping or recreating secondary indexes under the following circumstances:

- The InfoCube contains many uncompressed InfoPackages.
- The InfoCube contains no compressed InfoPackages
- The amount of data in the uncompressed F fact table is much higher than the amount of data in the compressed E fact table.

In single cases, there has been an overall improvement of performance for DELETE statements of up to 60 percent.



We recommend that you use a process chain to drop the secondary indexes before deletion or compression of InfoPackages and to recreate them afterwards.



You should not drop the indexes if the compressed E fact table is large or if the dataset for deletion or compression is small.

6.4 Increasing the Buffer Pool Hit Ratio

Concern

The hit ratio of a buffer pool is less than 97% and the performance of read or write operations is low.

Goal

To reduce the number of data pages that are released from the buffer pool.

Solution

Facts

If new data is read and no space is left in the buffer pool, other data pages are released from the buffer pool. As a consequence a lot of fluctuating data, which is read only once, can displace stable data that is read frequently and should reside in the buffer pool. Thus, the buffer pool hit ratio is reduced.

In a SAP BW installation, two default buffer pools are created that contain BW-specific data. The tablespace of master data tables and the tablespace of dimension tables are assigned to the same buffer pool as the SAP base tablespaces. Typically, master data and dimension data do not vary much in size and nature. They are read very frequently and should therefore always reside in the buffer pool.

The tablespaces for fact data, ODS tables, and PSA tables are assigned to the default buffer pool `BP_BW_16K`. Typically, this kind of data is large in size and individual data pages are not read very often. Data pages of fact, ODS, or PSA tables are released from time to time because normally many data pages are read and not all data fits at once into the buffer pool.



The reason for using a separate dimension buffer pool in the default installation is to prevent frequently used dimension data pages from being replaced in the buffer pool by fact data pages that are read only once.

Recommendation

To reduce the number of released data pages, you can either increase the size of the buffer pool or create separate buffer pools for the different kinds of data.

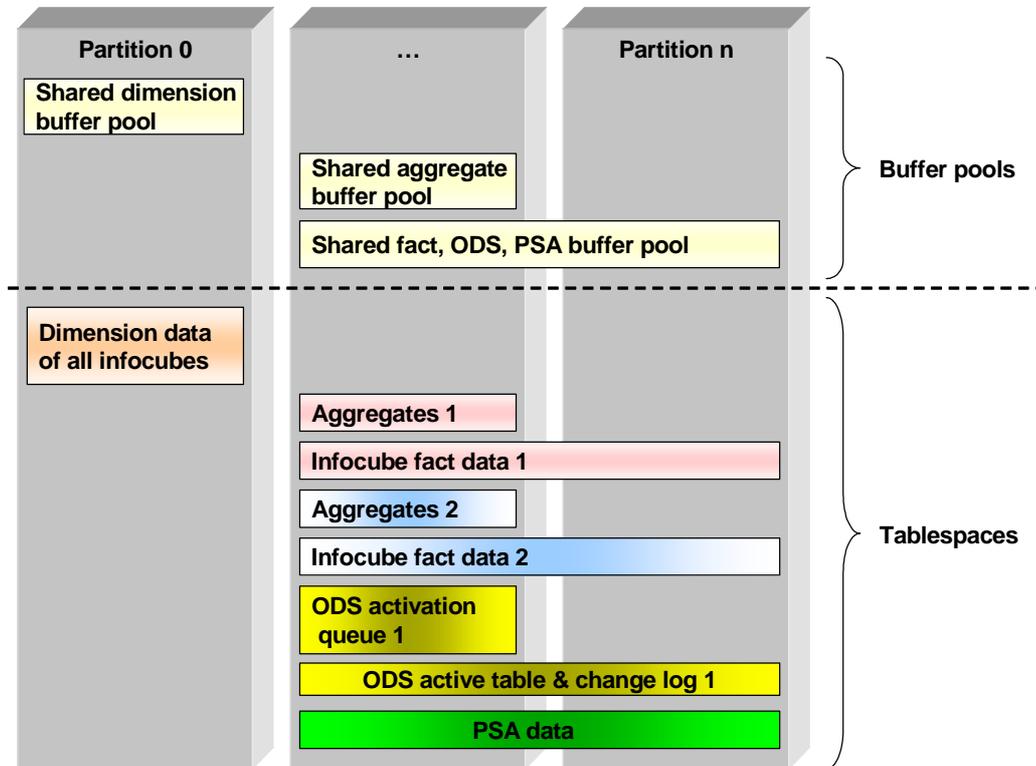
The amount of fluctuating data that is read at once (for example, fact data of an InfoCube read in a single query) can exceed the available main memory. In this case, you should create separate buffer pools for stable data and thereby prevent this stable data from being released from the buffer pool.

For more information, see [Creating Buffer Pools \[Page 28\]](#).

By default, the fact tables of aggregates are created in the same tablespace as the fact tables of the corresponding InfoCube. Hence, data pages with aggregate data are written to the same buffer pool as the data pages with the InfoCube's fact data. Aggregates are smaller than the InfoCubes' fact tables and they are read more frequently than InfoCube's fact tables. Similar to dimension data, you should create a separate buffer pool for aggregates. To do so, you create a tablespace and a buffer pool for aggregates and assign the new data class to the aggregates.

Example

The following graphic shows an example of how you can distribute buffer pools:



A separate buffer pool for dimension data is created that is different from the default dimension buffer pool `IBMDEFAULTBP` or `BP_STD_16K`. The buffer pool is not distributed across partitions and created on the same partition as the dimension tablespace. It is shared by all dimension tables of all InfoCubes.

The data of all InfoCube fact tables, ODS tables, and PSA tables is stored in tablespaces that are distributed across partitions 1 to n. These tablespaces are assigned to a buffer pool that is also created on partitions 1 to n.



The tablespace of the ODS activation queue table is created on a single partition but is also assigned to the multi-partition buffer pool.

The aggregate tablespace is created on a single partition other than partition 0. The corresponding separate aggregate buffer pool is created on the same partition. The buffer pool is shared by all aggregates of all InfoCubes.

6.5 Using Temporary Tablespaces

Concern

In an SAP BW system, temporary tablespaces can temporarily grow large in size. In particular when creating aggregates of large InfoCubes, the temporary tablespaces can grow up to more than 20 GB.

Goal

To choose a tablespace layout that achieves optimal performance.

Solution

In a SAP BW system, there are data tablespaces of a page size of 4 KB and 16 KB. Therefore, you need at least the temporary tablespaces `PSAPTEMP` and `PSAPTEMP16`. These temporary tablespaces should be created as SMS tablespaces, which is the default setting for an SAP system installation.

You can create the temporary tablespaces (`PSAPTEMP` and `PSAPTEMP16`) in the same file system (UNIX) or directory (Windows). However, you must make sure that enough free space is available to store all data. The total amount of free space depends on the size of the InfoCube.

If you are using a storage system, it is usually sufficient to plan one file system or directory for the temporary tablespaces. If you are not using a storage system, you can avoid I/O bottlenecks by distributing several file systems or directories of the temporary tablespaces over several physical disks. Thus, you can apply DB2's disk striping function.



You **cannot** distribute SMS tablespaces over additional file systems or directories after they have been created. If you plan to change the layout of the temporary tablespaces after installation, you should create an additional temporary tablespace with the same page size and the required layout. Then, you can drop the old tablespace.



There should **only** exist one temporary tablespace for each page size.

7 Additional Sources of Information

The following list provides additional sources of information:

- For detailed information about SAP BW, see SAP Service Marketplace at service.sap.com/bw
- For the installation of an SAP BW system on DB2 UDB for UNIX and Windows, see documentation *Installation Guide – SAP Web Application Server ABAP 6.40 on <Operating system platform>: IBM DB2 Universal Database for UNIX and Windows*.
You can find this documentation in SAP Service Marketplace at service.sap.com/instguidesNW04 → *Installation*.
- For information about how to build and scale an SAP BW system, see the IBM redbook *Building and Scaling an SAP Business Information Warehouse on DB2 UDB ESE*, no. SG24-7094.
You can download this redbook as a PDF at www.ibm.com/redbooks.
- For information about the released operating systems for an SAP BW installation, see the *Product Availability Matrix (PAM)* in SAP Service Marketplace at service.sap.com/pam.
- SAP Notes

Note Number	Title
609384	Known Errors and Available Fixes in DB2 UDB V8
584952	DB2 UDB ESE V81 Standard Parameter Settings
546262	DB6: Performance on SAP BW 30B, BW 31 Content and BW 3.5
374502	DB6: DB2 UDB ESE BW Performance Overview of notes
192658	Setting basis parameters for BW Systems