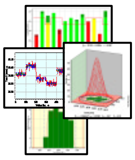


Q-DAS Database Manual

Oracle Instances and Scripts
Maintenance Procedure

Q-DAS Library
Living Documentation
Last edited: 09/13/08

Version: EH

q-das@q-das.com *Tel. 248-299-4910*

Table of Contents

1 Oracle Database Versions

Q-DAS software runs with Oracle databases of version 7.3 or higher. This will probably also apply to the most recent versions but might not yet have been confirmed by Q-DAS.

When **Oracle 10g** is used, it is **highly recommended** to have a close look at the sub version number: If the number is 10.0.2 or lower, **it must be updated at least to version 10.0.3** to avoid a severe bug which may occur when working with different schemas in the same database instance.

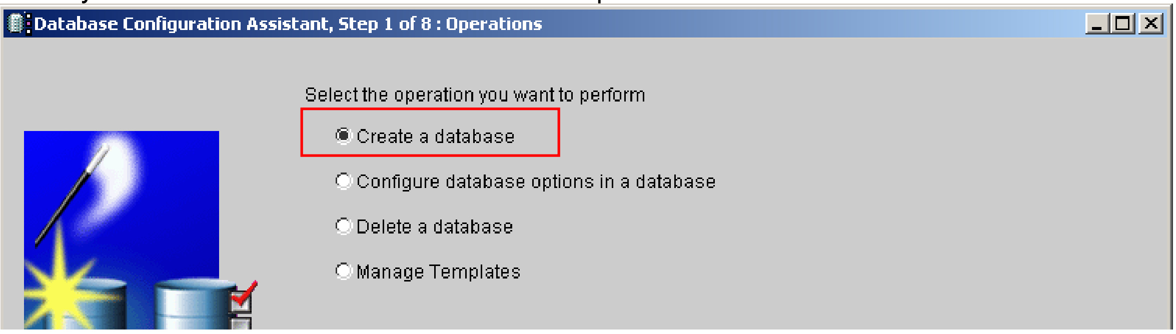
2 Creating a New Database

2.1 Creating a New Database Instance

For creating a new database instance, you need the name of the instance (e.g. BLOCKDB). For Oracle, use

Programs
    Oracle – OraHomeXX
        Configuration and Migration Tools
            Database Configuration Assistant

There you can select the “Create a database” option and continue with the “Next” button.



Database Configuration Assistant, Step 1 of 8 : Operations

Select the operation you want to perform

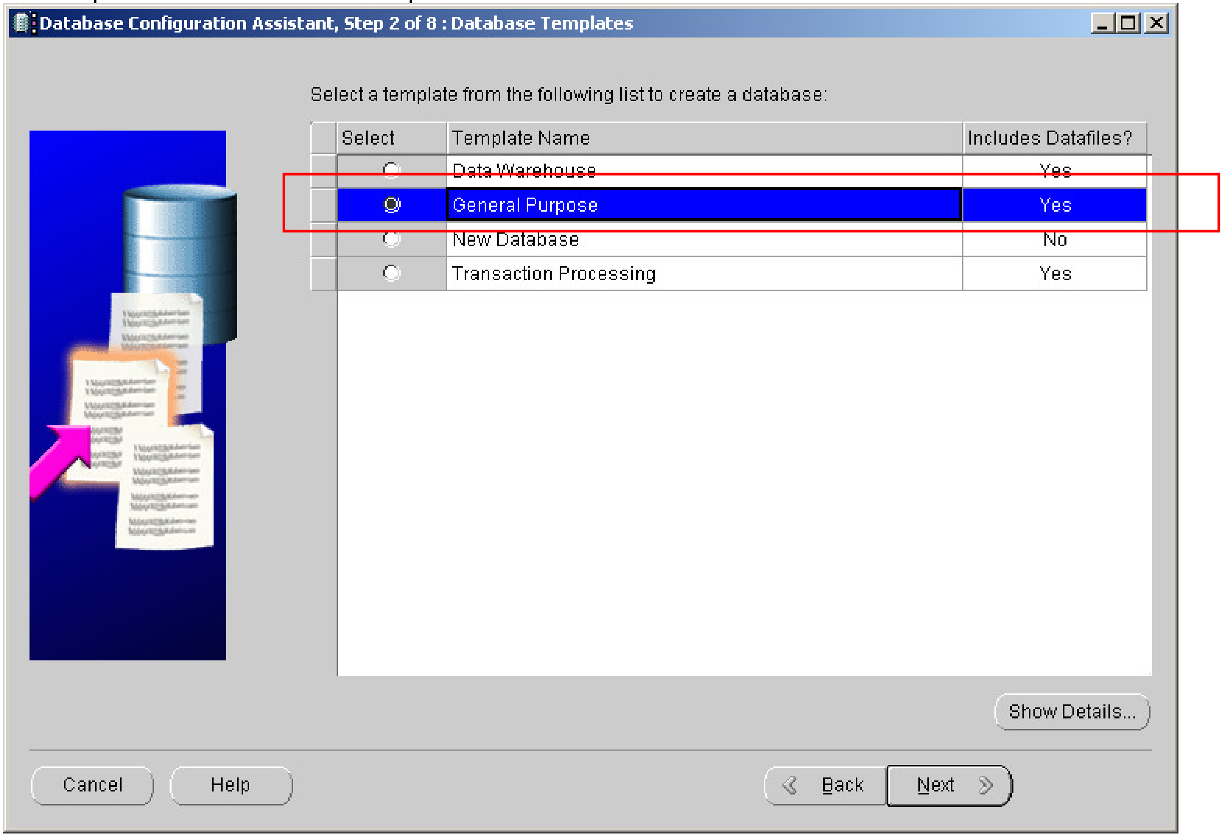
● Create a database

○ Configure database options in a database

○ Delete a database

○ Manage Templates

As template select “General Purpose”

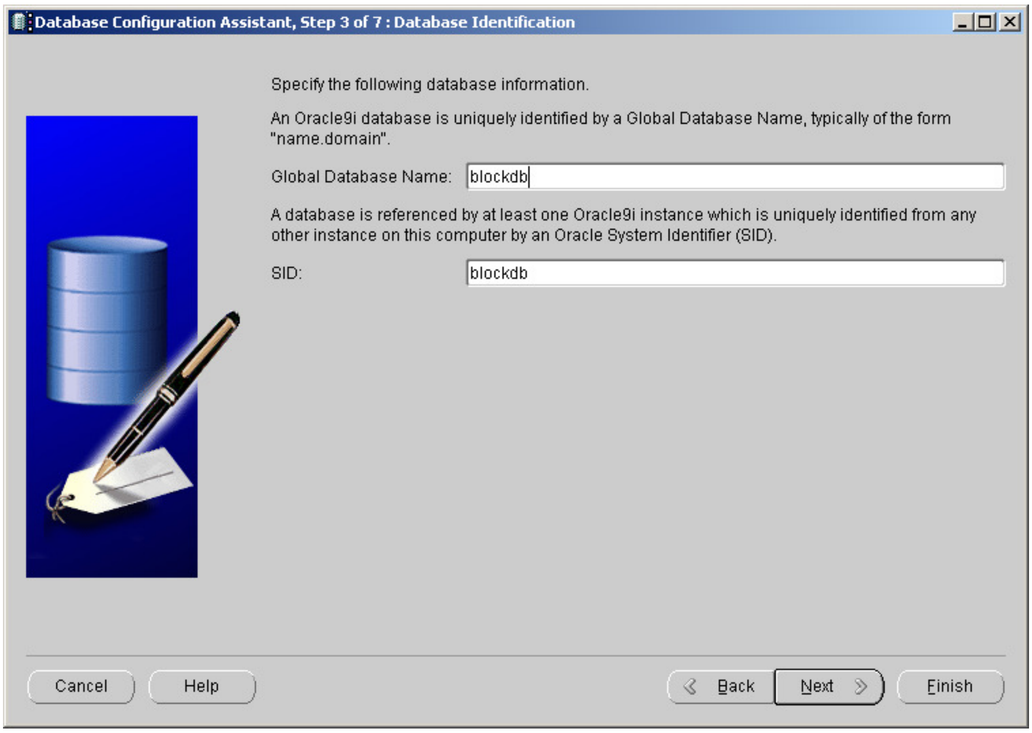


Database Configuration Assistant, Step 2 of 8 : Database Templates

Select a template from the following list to create a database:

Select	Template Name	Includes Datafiles?
○	Data Warehouse	Yes
●	General Purpose	Yes
○	New Database	No
○	Transaction Processing	Yes

Define a **global database name** e.g. BLOCKDB, as well as, the **SID** e.g. BLOCKDB
Note: It is recommended that these be named the same.


Database Configuration Assistant, Step 3 of 7 : Database Identification

Specify the following database information.

An Oracle9i database is uniquely identified by a Global Database Name, typically of the form "name.domain".

Global Database Name: blockdb

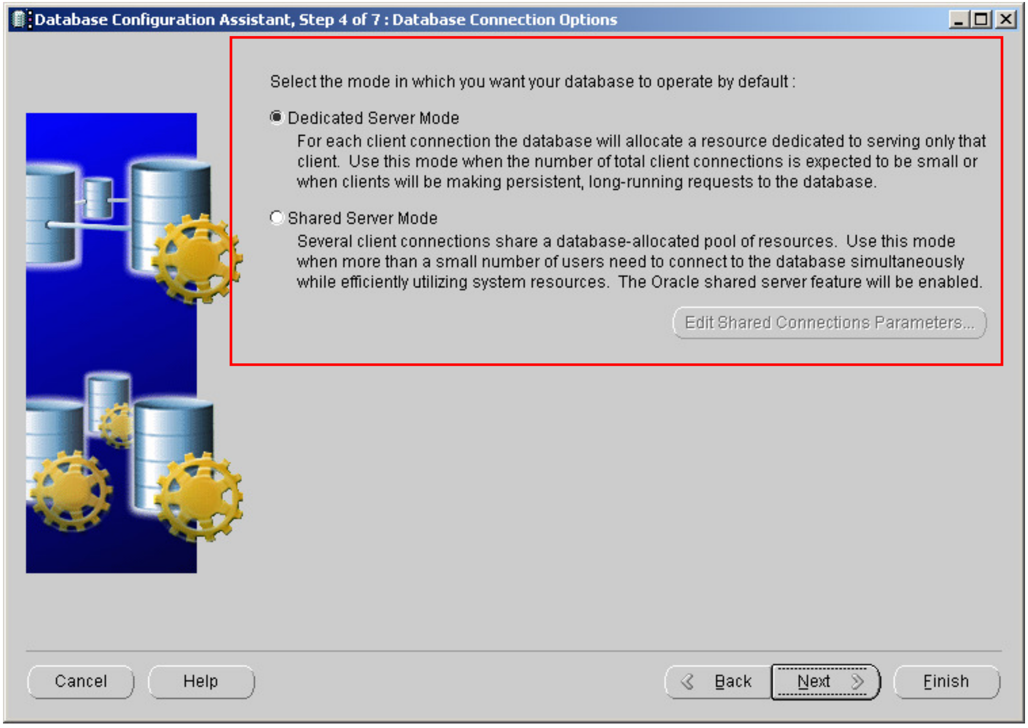
A database is referenced by at least one Oracle9i instance which is uniquely identified from any other instance on this computer by an Oracle System Identifier (SID).

SID: blockdb

Cancel
Help
Back
Next
Finish

Click on “Next” to proceed

Depending on the number of user the mode in which the users will be working with the database, define either the “**Dedicated**” **or** the “**Shared**” Server Mode to be used.

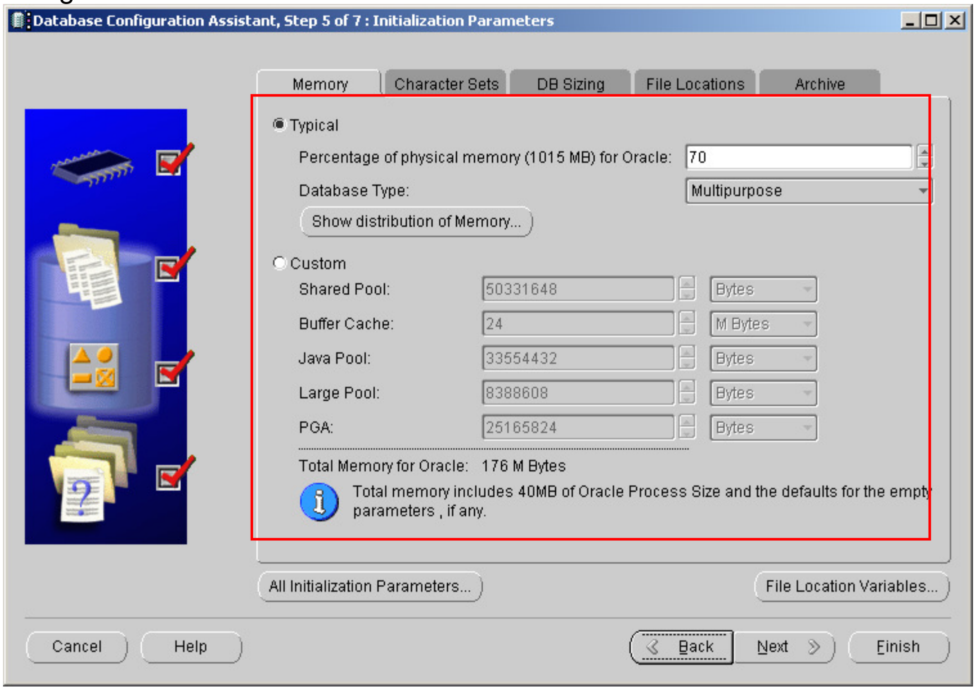

Database Configuration Assistant, Step 4 of 7 : Database Connection Options

Select the mode in which you want your database to operate by default :

● Dedicated Server Mode
For each client connection the database will allocate a resource dedicated to serving only that client. Use this mode when the number of total client connections is expected to be small or when clients will be making persistent, long-running requests to the database.

○ Shared Server Mode
Several client connections share a database-allocated pool of resources. Use this mode when more than a small number of users need to connect to the database simultaneously while efficiently utilizing system resources. The Oracle shared server feature will be enabled.

Either select the “**Typical**” Memory distribution and enter the desired percentage as well as select “Multipurpose” from the pull-down-menu, **or** pick the “**Custom**” option and make the required settings.



Database Configuration Assistant, Step 5 of 7 : Initialization Parameters

● Typical

Percentage of physical memory (1015 MB) for Oracle: 70

Database Type: Multipurpose

○ Custom

Shared Pool: 50331648 Bytes

Buffer Cache: 24 M Bytes

Java Pool: 33554432 Bytes

Large Pool: 8388608 Bytes

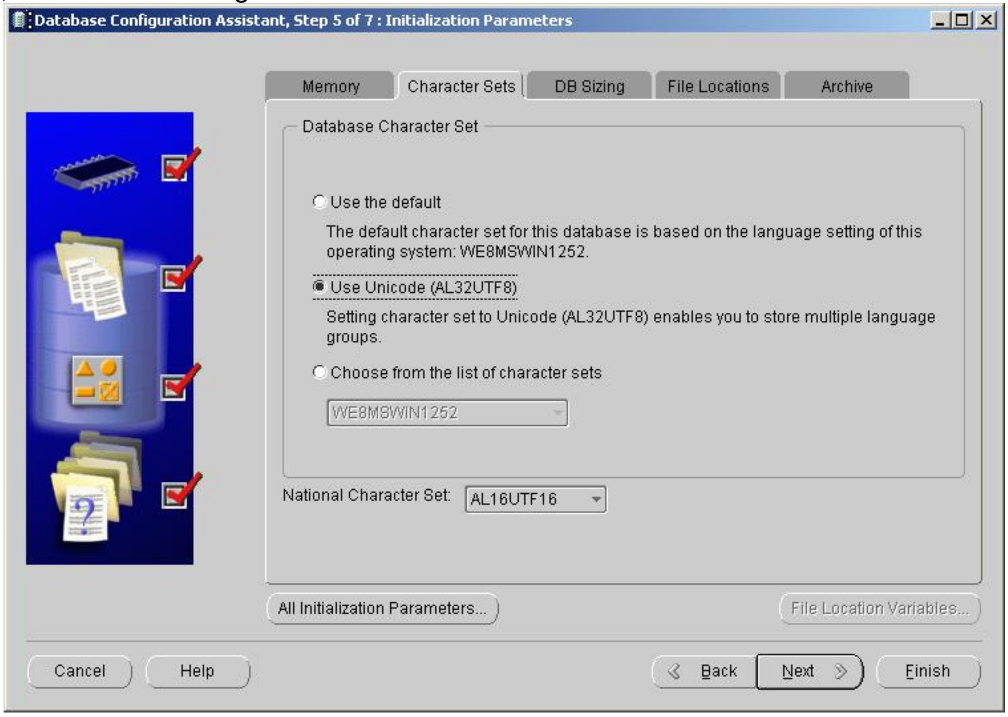
PGA: 25165824 Bytes

Total Memory for Oracle: 176 M Bytes

Total memory includes 40MB of Oracle Process Size and the defaults for the empty parameters , if any.

Memory
Character Sets
DB Sizing
File Locations
Archive
Show distribution of Memory...
All Initialization Parameters...
File Location Variables...
Cancel
Help
Back
Next
Finish

In case of the necessity of using Unicode characters, e.g. with **Asian languages**, the database instance should use **Unicode characters (AL32UTF8)** instead of the standard Western European character set. This setting has to be done during the creation of the database instance, it cannot be changed once the instance has been created.



Database Configuration Assistant, Step 5 of 7 : Initialization Parameters

Database Character Set

○ Use the default

The default character set for this database is based on the language setting of this operating system: WE8MSWIN1252.

● Use Unicode (AL32UTF8)

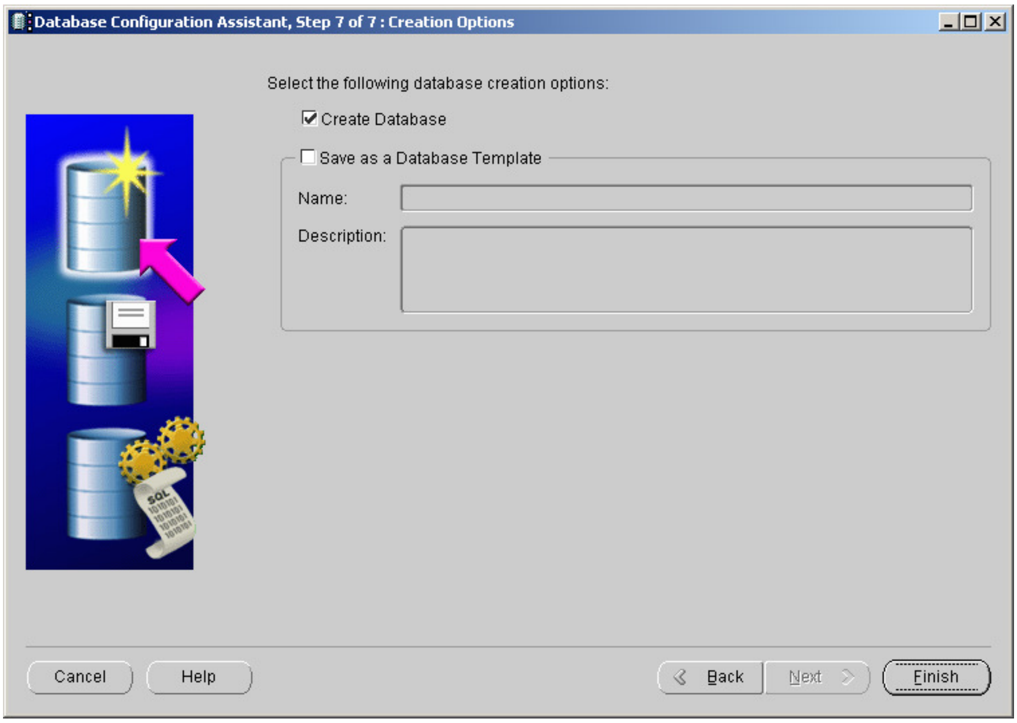
Setting character set to Unicode (AL32UTF8) enables you to store multiple language groups.

○ Choose from the list of character sets

WE8MSWIN1252

National Character Set: AL16UTF16

Click on “Finish” to create the database instance. A second window will pop up which allows you to double check all settings made. If all settings are correct, click on “Next” to continue and start thee database creation with “Finish”.



Database Configuration Assistant, Step 7 of 7 : Creation Options

Select the following database creation options:

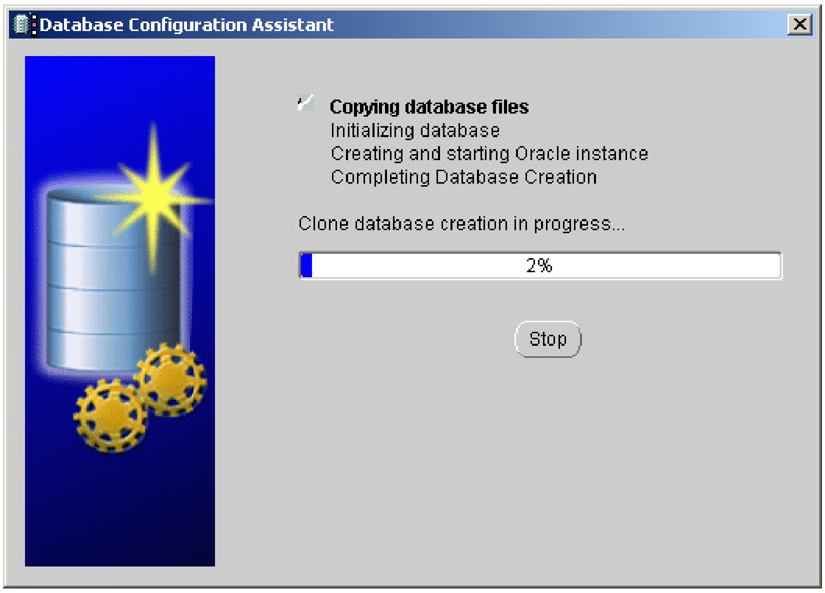
☑ Create Database

□ Save as a Database Template

Name:

Description:

The database creation may take a little while. Please allow sufficient time for the Database Assistant to finish.



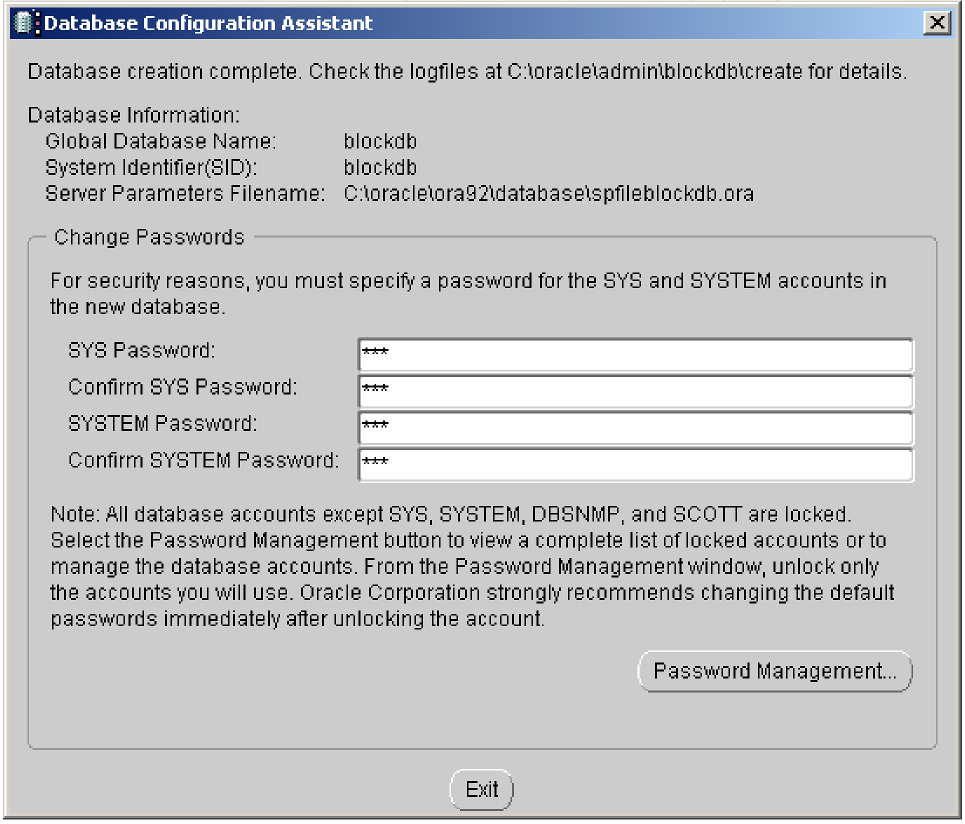
Database Configuration Assistant

✓ **Copying database files**
Initializing database
Creating and starting Oracle instance
Completing Database Creation

Clone database creation in progress...

2%

It may be required to enter new passwords for the Sys and the System account for the new database. Please do so, if a respective dialog window pops up.



Database Configuration Assistant

Database creation complete. Check the logfiles at C:\oracle\admin\blockdb\create for details.

Database Information:

Global Database Name:	blockdb
System Identifier(SID):	blockdb
Server Parameters Filename:	C:\oracle\ora92\database\spfileblockdb.ora

Change Passwords

For security reasons, you must specify a password for the SYS and SYSTEM accounts in the new database.

SYS Password:	***
Confirm SYS Password:	***
SYSTEM Password:	***
Confirm SYSTEM Password:	***

Note: All database accounts except SYS, SYSTEM, DBSNMP, and SCOTT are locked. Select the Password Management button to view a complete list of locked accounts or to manage the database accounts. From the Password Management window, unlock only the accounts you will use. Oracle Corporation strongly recommends changing the default passwords immediately after unlocking the account.

2.2 Configuring the Database Instance

Get the name of the tablespaces with the following command under SQL Plus:

```
SELECT TABLESPACE_NAME FROM DBA_TABLESPACES;
```

Write down the names of the tablespaces for the temporary data, the user data and the rollback data. (Oracle 8.i : e. g.: temp, users, rbs)

2.2.1 Script to initialize the tablespaces: “init_db.sql”, “init_tablespaces.sql” or “init_tablespaces_9i.sql”

The script used to create the tablespace(s) will determine the script modifications needed to be made.

All modifications of the script files, should be saved in a directory called

e.g. \ORAXX\ADMIN\BLOCKDB\SCRIPTS, **so that the changes can be reproduced.**

Different scripts are available on the CD, depending on the Oracle Version used.

Script name	Oracle Version
init_db.sql	Oracle 7
init_tablespaces.sql	Oracle 8
init_tablespaces_9i.sql	Oracle 9i or higher

No matter what Oracle Version is used, the script has to be adjusted to the present directory structure, tablespace names and desired initial file size.

2.2.1.1 Required adjustments in any init_tablesspaces script

Modify the script “init_tablespaces.sql” with the names of the selected tablespaces.
(e.g. temporary_data will be temp)

You are also able to set the size of the new QDAS database, the number of data files and the directory where the database should be located. In the script are the following entries:

```
e.g.:   create tablespace qwert_table
        datafile '<pathname\withinstancename>\qwert_tab.dbf' size 10M reuse
        autoextend on next 10 M maxsize unlimited
```

Create subdirectories for each instance of the database on the hard disk, where the tablespaces should reside.

e.g. C:\DATA\BLOCKDB for the datafiles and

D:\INDEX\BLOCKDB for the index files

In the script, specify the location and initial size of the QDAS_DEFAULT data files (contains miscellaneous Q-DAS data)

e.g. create tablespace qdas_default
datafile **'C:\DATA\BLOCKDB\ qdas_default.dbf'** size **10M** reuse
autoextend on next 10 M maxsize unlimited

In the script, specify the location and initial size of the QWERT_TAB data files (table for measurement values)

e.g. create tablespace qwert_table
datafile **'C:\DATA\BLOCKDB\qwert_tab.dbf'** size **100M** reuse
autoextend on next 10 M maxsize unlimited

In the script, specify the location and initial size of the QMERK_TAB data files (table for characteristic and part header information)

e.g. create tablespace qmerk_table
datafile **'C:\DATA\BLOCKDB\qmerk_tab.dbf'** size **10M** reuse
autoextend on next 10 M maxsize unlimited

In the script, specify the location and initial size of the QWERT_IND data files (indices for measurement values)

e.g. create tablespace qwert_index
datafile **'D:\INDEX\BLOCKDB\qwert_ind.dbf'** size **100M** reuse
autoextend on next 10 M maxsize unlimited

In the script, specify the location and initial size of the QMERK_IND data files (indices for characteristic and part header information)

e.g. create tablespace qwert_index
datafile **'D:\INDEX\BLOCKDB\qmerk_ind.dbf'** size **10M** reuse
autoextend on next 10 M maxsize unlimited

Save and run the appropriate script with an Oracle DBA account in SQL Plus with the command:

```
START <Path>\<script name>.sql;
```

2.2.1.2 Additional adjustments for Oracle 7 (script “init_db.sql”)

In Oracle 7, the user_data tablespace and rollback segments are used. The initial storage size of the tablespaces for user_data (e.g. users) and for the rollback segments has to be adjusted, as well as the size and number of the rollback segments themselves:

In the script, set the initial storage size for the users_data and the rollback tablespace. Adjust the tablespace names as required.

e.g.
```
alter tablespace user_data
default storage(initial 100K next 500K);
alter tablespace rbs
default storage(initial 5M next 5M);
```

In the script, set the individual rollback segments to extent to unlimited (RBS1..RBS24 or RB1..RB24). Add more rollback segments if required. First, all rollback segments are set offline, then the extension to unlimited is set and afterwards the segments are set back online.

e.g.
```
alter rollback segment rbs1 offline;
alter rollback segment rbs1 storage (NEXT 5M MAXEXTENTS UNLIMITED);
alter rollback segment rbs1 online;
```

etc.

In this script, the default storage size for all QDAS tablespaces is set with the additional command for each QDAS tablespace.
Adjust the default storage size in this command line where required for Oracle 7:

e.g.
```
default storage(initial 10 K next 50 K maxextents unlimited);
```

2.2.1.3 Additional adjustments for Oracle 8 (script “init_tablespaces.sql”)

In Oracle 8, the rollback segments are used. The initial storage size of the tablespaces for the rollback segments (e.g. rbs) has to be adjusted, as well as the size and number of the rollback segments themselves:

In the script, set the initial storage size for the rollback tablespace. Adjust the tablespace names as required.

```
e.g.    alter tablespace rbs
        default storage(initial 5M next 5M);
```

In the script, set the individual rollback segments to extent to unlimited (RBS1..RBS24 or RB1..RB24). Add more rollback segments if required. First, all rollback segments are set offline, then the extension to unlimited is set and afterwards the segments are set back online.

```
e.g.    alter rollback segment rbs1 offline;
        alter rollback segment rbs1 storage (NEXT 5M MAXEXTENTS UNLIMITED);
        alter rollback segment rbs1 online;
```

etc.

In this script, the Extent management for all QDAS tablespaces is set to the following default with the additional command for each QDAS tablespace for Oracle 8:

```
EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
```

2.2.1.4 Additional adjustments for Oracle 9i or higher (script “init_tablespaces_9.sql”)

The size of the temporary tablespace may need to be adjusted manually in Oracle Enterprise Manager if the command in the script fails. Recommendation is to set the size for the temporary tablespace to “unlimited” (32 GB) where the Operating System permits.

In this script, the Extent management and Segment space management for all QDAS tablespaces is set to the following default with the additional command for each QDAS tablespace for Oracle 9i:

```
EXTENT MANAGEMENT LOCAL NOLOGGING
SEGMENT SPACE MANAGEMENT AUTO;
```

2.2.2 Scripts to initialize the user schema: “cre_user.sql” or “CRE_USER_10g.SQL”

This script creates the user for the database (e.g. qdas) and relates the default tablespaces. The user will be the table owner and be used to establish the database connection from the qs-STAT computers to the database.

Using the standard script, the user will be called QDAS with the password QDAS.

Different scripts are available on the CD, depending on the Oracle Version used.

Script name	Oracle Version
CRE_USER.sql	Oracle 7, 8 and 9i
CRE_USER_10g.sql	Oracle 10g and higher

The CRE_USER script contains the following commands. Adjust the names of the tablespaces if it is necessary:

```
create USER QDAS IDENTIFIED BY QDAS;
grant connect to QDAS;
grant resource to QDAS;
alter user QDAS default tablespace qdas_default;
alter user QDAS temporary tablespace temp;
```

Due to changed default permissions in Oracle 10g, the CRE_USER_10g script contains the following additional commands:

```
grant create view to QDAS;
alter user QDAS QUOTA UNLIMITED ON qdas_default;
alter user QDAS QUOTA UNLIMITED ON qwert_table;
alter user QDAS QUOTA UNLIMITED ON qmerk_table;
alter user QDAS QUOTA UNLIMITED ON qwert_index;
alter user QDAS QUOTA UNLIMITED ON qmerk_index;
```

Note: It is possible to use a different user name (and of course a different password). In this case the user name “QDAS” has to be replaced by the desired name in every command of the CRE_USER or CRE_USER_10g script, and the password “QDAS” in the “IDENTIFIED BY” clause has to be replaced by the desired password.

Save and run the appropriate script with an Oracle DBA account in SQL Plus with the command:

```
START <Path>\<script name>.sql;
```

3 Installing the Q-DAS Tables

3.1 Q-DAS Tables (all Me x versions)

If you want to install the tables for **Q-DAS Me x** on a fresh instance with the standard Q-DAS database setup**,** run **only** the script **Tables_2000.sql**. Please always use the current scripts from the CD as the scripts are updated for each new version.

3.1.1 Connect with the appropriate Oracle user

All scripts for generating the tables and database architecture have to be run with the newly created user schema (default: QDAS), for it to become the table owner.

Connect to the user "QDAS" with the command in SQL Plus:

```
Connect QDAS/QDAS@QDASDB;
(QDASDB is the database alias, it may be different)
```

3.1.2 Script "Tables_2000.sql":

To create the tables necessary for using qs-STAT or procella, use the script "Tables_2000.sql".

Run this script with the command:

```
START <Path>\tables_2000.sql;
```

After that, you can quit the ***sql plus*** window by using the command "**exit**".

The database creation of the standard QDAS database is now completed.

3.1.3 Additional Customized scripts (where applicable)

3.1.3.1 In general

For performance improvement, it may be useful that not all fields in the parts table TEIL and characteristics table MERKMAL have to be created.
It is possible to modify the tables accordingly with an additional script so that not all K-fields are created.
However, all edits and customizations to the scripts for table creation in the QDAS database should be performed by a Q-DAS representative.

In table TEIL the following columns must not be deleted:
TETEIL and the columns used in the upload as key fields.

In table MERKMAL the following columns must not be deleted:
METEIL
MEMERKMAL
MEZYKLUS
MEGLEITSTUMF
MEPRUEFZST
MEAUGROUP
MEUPPERMERKMAL
MEPRUEFORT
MEUNTERSART
MEPMGRUPPE
and the columns used in the upload as key fields.

3.1.3.2 Customized Scripts for GMPT

3.1.3.2.1 “Modify_existingcharacteristics_GM.sql”

This script has to be run if you want to reduce the size of the characteristic table to the GM K-fields. This script drops some additional data fields in the characteristic table which are implemented in the standard 16-bit and ME x scripts.

After creating the standard database with the “Tables_2000.sql” script, connect to the user “QDAS” with the command in SQL Plus:

```
Connect QDAS/QDAS@QDASDB;
```

(QDASDB is the database alias, it may be different within your facility)

To modify the characteristics table, use the script “Modify_existingcharacteristics_GM.sql”.

Run this script with the command:

```
START <Path>\Modify_existingcharacteristics_GM.sql;
```

3.1.3.2.2 Script “Tables_upgrade_2000special.sql”:

For any **Q-DAS millennium (all Me x) database version**, or for upgrading from special versions of qs-stat 3.x to qs-stat millennium, run the **“Tables_upgrade_2000special.sql” after the “Modify_existingcharacteristics_GM.sql”** script:

Connect to the user “QDAS” with the command in SQL Plus:

```
Connect QDAS/QDAS@QDASDB;
```

(QDASDB is the database alias, it may be different within your facility)

After that, you can quit the ***sql plus*** window by using the command “**exit**”.

3.1.3.3 Customized Script for Ford Europe

3.1.3.3.1 “Modify_existingtables_Ford.sql”

This script has to be run if you want to reduce the size of the characteristic table to the Ford Europe K-fields. This script drops some additional data fields in the characteristic table which are implemented in the standard 16-bit and ME x scripts.

After creating the standard database with the “Tables_2000.sql” script, connect to the user “QDAS” with the command in SQL Plus:

```
Connect QDAS/QDAS@QDASDB;
(QDASDB is the database alias, it may be different within your facility)
```

To modify the characteristics table, use the script “Modify_existingcharacteristics_Ford.sql”.

Run this script with the command:

```
START <Path>\Modify_existingcharacteristics_Ford.sql;
```

3.1.3.3.2 Script “Tables_upgrade_2000special.sql”:

For any **Q-DAS millennium (all Me x) database version**, or for upgrading from special versions of qs-stat 3.x to qs-stat millennium, run the **“Tables_upgrade_2000special.sql” after the “Modify_existingcharacteristics_Ford.sql”** script:

Connect to the user “QDAS” with the command in SQL Plus:

```
Connect QDAS/QDAS@QDASDB;
(QDASDB is the database alias, it may be different within your facility)
```

After that, you can quit the ***sql plus*** window by using the command “**exit**”.

3.2 Updating a ME database from an older ME to a newer ME version

3.2.1 Preparation

Before updating an existing database:

- Make sure that you have a current backup
- Make sure that no user is connected to the database. This includes the database Upload program: the Upload must be stopped and closed.

3.2.2 Script “Tables_upgrade_2000.sql”:

Copy the “Tables_upgrade_2000.sql” script from the current CD.

Open a SQL Plus window and connect to the database with the user “QDAS” with the command:

```
Connect QDAS/QDAS@QDASDB;
```

(QDASDB is the database alias, it may be different)

Run the update script with the command:

```
START <Path>\tables_upgrade_2000.sql;
```

Depending on the version of the existing database, the script will generate errors notifying you that certain elements (tables, column etc.) already exist in the database.
These messages are normal as the script can be used to update any ME database version to the current version number.

After that, you can close the ***sql plus*** window by using the command “**exit**”.

The Database Upload can now be started up again and the connections to the database can be re-established.

4 Dropping the tables, user or the database instance

4.1 Introduction

Dropping the tables, user or database instance should only be done if you

- Do not have any data in the database
- You want to start over the initial installation and start from scratch

If there is already data in the database, any of the following procedure will erase the data for good!

4.2 Dropping the database tables

Open a SQL Plus window and connect to the database with the table owner user “QDAS” with the command:

```
Connect QDAS/QDAS@QDASDB;
```

(QDASDB is the database alias, it may be different)

For deleting tables for a **16-bit database**, run the “**DROPTABLES_16BIT.SQL**” script.
For deleting tables for a **ME database**, run the current “**DROPTABLES_2000.SQL**” script.

Run the appropriate script with the command:

```
START <Path>\<script name>;
```

4.3 Deleting the User

To delete the user **with all its related objects**, open a SQL Plus window and connect to the database **as dba** and call

```
DROP USER QDAS CASCADE;
```

For deleting the user specific tablespaces, in addition call the script
“**DROP_tablespaces.sql**”

The data files have to be deleted manually afterwards.

4.4 Deleting the instance

For deleting an entire database instance, you need the name of the instance (e.g. BLOCKDB).
For Oracle, use

```
START
        Programs
                OraHomeXX
                        Database Administration
                                Database Configuration Assistant
```

There you can select the database instance to be removed. This also removes all data files associated with the instance.

5 Export and Import a Database

5.1 Export a Database

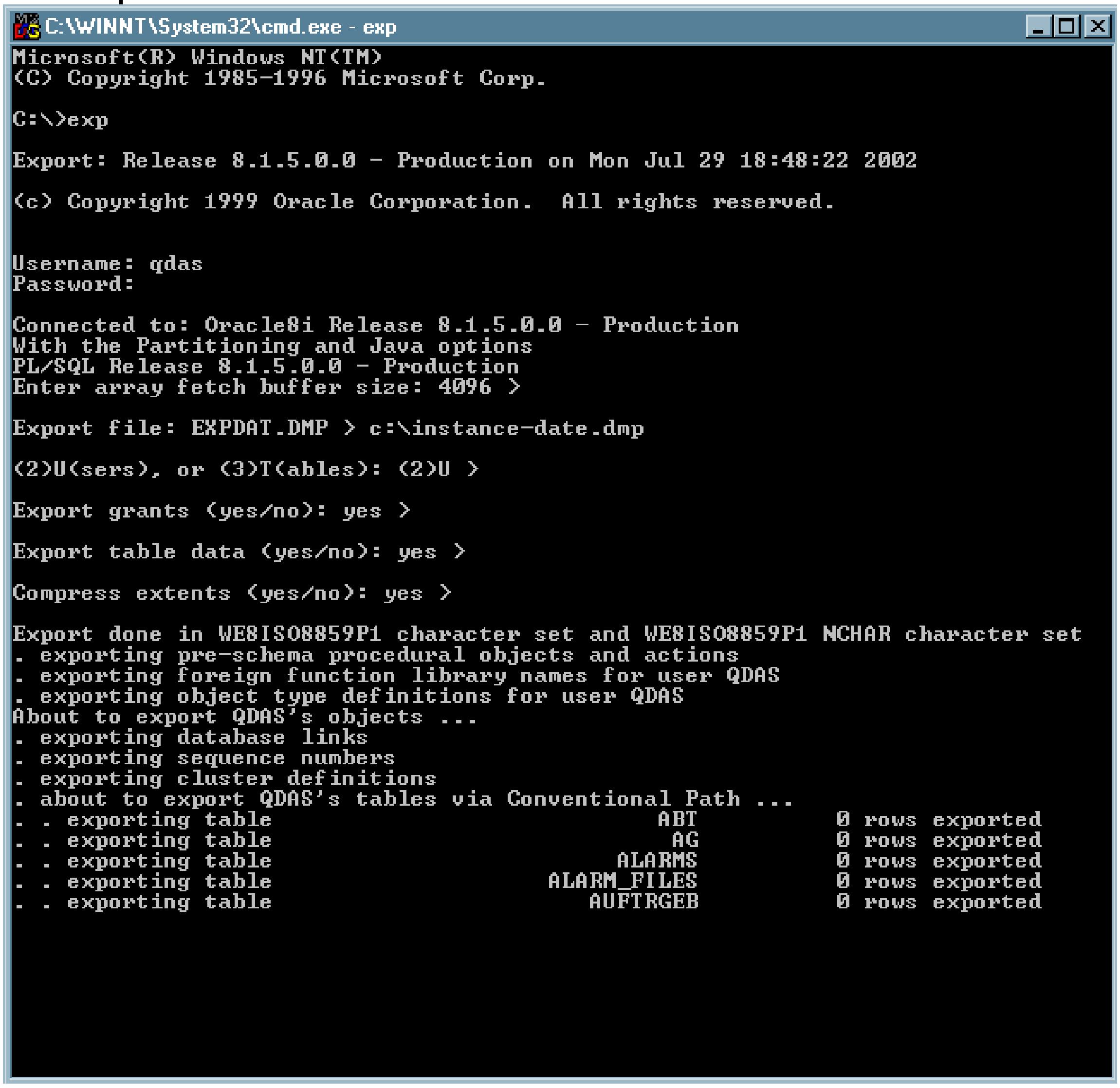
At the command line set the default oracle instance where the data should be imported to (only necessary, if more than one instance is installed).

SET ORACLE_SID=<INSTANCE_NAME>

e.g.

SET ORACLE_SID=BLOCKDB

Then call **exp.exe** from the command line:



```
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

C:\>exp

Export: Release 8.1.5.0.0 - Production on Mon Jul 29 18:48:22 2002

(c) Copyright 1999 Oracle Corporation.  All rights reserved.


Username: qdas
Password:

Connected to: Oracle8i Release 8.1.5.0.0 - Production
With the Partitioning and Java options
PL/SQL Release 8.1.5.0.0 - Production
Enter array fetch buffer size: 4096 >

Export file: EXPDAT.DMP > c:\instance-date.dmp

(2)U(sers), or (3)T(ables): (2)U >

Export grants (yes/no): yes >

Export table data (yes/no): yes >

Compress extents (yes/no): yes >

Export done in WE8ISO8859P1 character set and WE8ISO8859P1 NCHAR character set
. exporting pre-schema procedural objects and actions
. exporting foreign function library names for user QDAS
. exporting object type definitions for user QDAS
About to export QDAS's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export QDAS's tables via Conventional Path ...
. . exporting table                            ABT          0 rows exported
. . exporting table                             AG          0 rows exported
. . exporting table                         ALARMS          0 rows exported
. . exporting table                    ALARM_FILES          0 rows exported
. . exporting table                       AUFTRGEB          0 rows exported
```

5.2 Import a Database

At the command line set the default oracle instance where the data should be exported from (only necessary if more than one instance is installed).

SET ORACLE_SID=<INSTANCE_NAME>

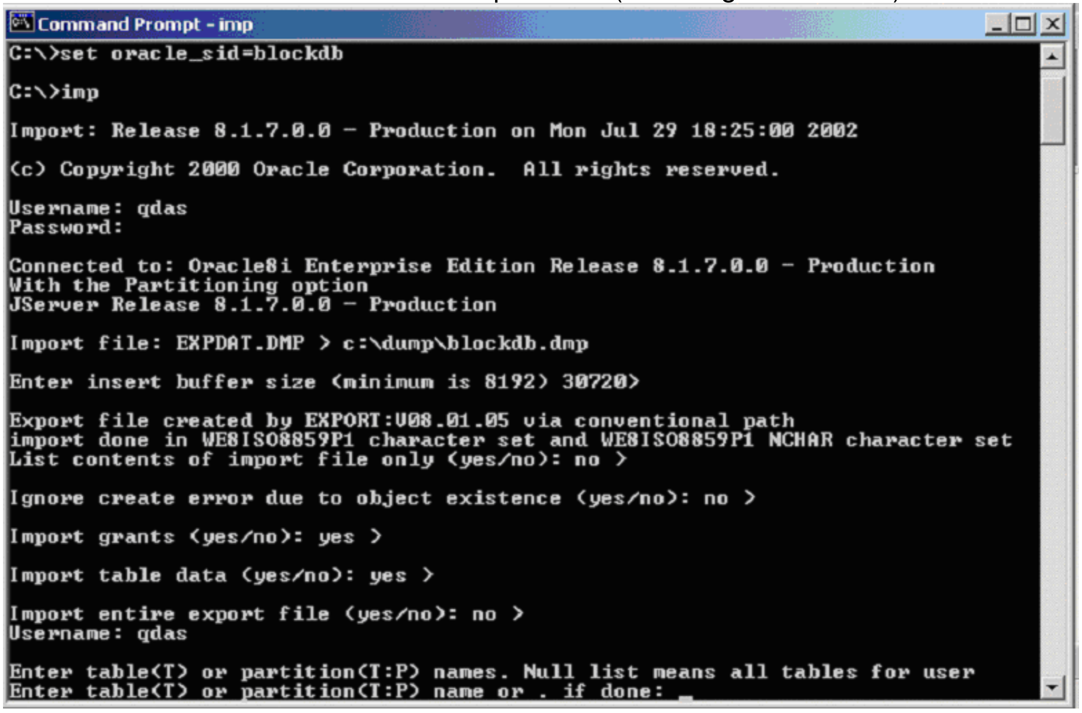
e.g.

SET ORACLE_SID=BLOCKDB

The tables for the imported user must not exist but the tablespaces and the user itself must exist already.

Then call **imp.exe** from the command line:

Logon using qdas, password qdas
Import file- specify path of the file to be imported
Press enter for the remainder of the questions (choosing the defaults)



```
C:\>set oracle_sid=blockdb

C:\>imp

Import: Release 8.1.7.0.0 - Production on Mon Jul 29 18:25:00 2002

(c) Copyright 2000 Oracle Corporation.  All rights reserved.

Username: qdas
Password:

Connected to: Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
With the Partitioning option
JServer Release 8.1.7.0.0 - Production

Import file: EXPDAT.DMP > c:\dump\blockdb.dmp

Enter insert buffer size (minimum is 8192) 30720>

Export file created by EXPORT:V08.01.05 via conventional path
import done in WE8ISO8859P1 character set and WE8ISO8859P1 NCHAR character set
List contents of import file only (yes/no): no >

Ignore create error due to object existence (yes/no): no >

Import grants (yes/no): yes >

Import table data (yes/no): yes >

Import entire export file (yes/no): no >
Username: qdas

Enter table(T) or partition(T:P) names. Null list means all tables for user
Enter table(T) or partition(T:P) name or . if done: _
```

6 Database Maintenance Procedure

In addition to daily/weekly monitoring of the database system, a monthly maintenance procedure must be followed. This maintenance procedure will help with diagnosing database performance degradation.

All maintenance history, should be saved in a directory called

e.g. \ORA81\ADMIN\<instance name>\MAINTENANCE\

In this maintenance directory, there should be two folders: DBINFO and TIME-TESTS. These two folders will hold the results of the monthly maintenance procedure.

Note: This maintenance procedure must be followed for EACH database instance.

6.1 Export of the Database

For each database instance, create an export. This export should be kept in the Admin directory, noting the date that the export was created.

6.2 Information about Database Contents

For each instance, content information must be collected. Two scripts will help collect this data: **DBINFO.sql and PART_CHAR_VALUESINFO.sql**.

6.2.1 General Info about the Database

The script “dbinfo.sql” will collect information about the database structure. This script will create a text file named “dbinfo.txt” in the system directory.
Connect to the user QDAS/QDAS@<instance name>, using *sql Plus*.
To run this script,

```
START <Path>\dbinfo.sql
```

Once the script is finished, rename the created text file to include the date in which the file was created.

e.g. 07-31-02-dbinfo.txt

Place this file in the **DBINFO directory** for the related instance.
(See Index for example of dbinfo.sql output.)

5.2.2 Info about Parts, Characteristics and Values

The script “part_char_valuesinfo.sql” will collect information about the amount of data in the database instance. This script will create a text file named “spool.txt” in the C:\TEMP directory.
Connect to the user QDAS/QDAS@<instance name>, using *sql Plus*.
To run this script,

```
START <Path>\part_char_valuesinfo.sql
```

Once the script is finished, rename the created text file to include the date in which the file was created.

e.g. 07-31-02-spool.txt

Place this file in the **DBINFO directory** for the related instance.
(See Index for example of part_char_valuesinfo.sql output.)

6.3 Time Testing with Reference Data

Time tests should be performed on each database instance. Over the life of the database, the documented time tests will help with grading the database.

5.3.1 Test Part

Predefined test parts will be used for half of the time tests. These test parts will include a part containing few characteristics with many values per characteristic, and a part containing many characteristics with a single value per characteristic.

Procedure:

1. **Upload each part separately**. Record the time it takes for each upload. Also record the Upload PC specifications. (CPU, memory, etc.)
2. **Query each part separately**, from the same client machine. Record the time it takes for each query. Also record the client PC specifications. (CPU, memory, etc.)
3. **Record these times in a text file**. Name the text file “xx-xx-xx-test-part.txt”, including the date in which the tests were taken.
4. Place this file in the time-test folder, located in the maintenance directory.

5.3.2 Actual Data

Actual data will be used for the other half of the time tests. Two existing parts from the database will be used, each representing the type of parts used for in the previous time test. (a part containing few characteristics with many values per characteristic, and a part containing many characteristics with a single value per characteristic)

Note: To collect this data, stop the upload process and collect some relevant files waiting to be uploaded. This will assure that the data does not get duplicated in the database.

Procedure:

1. **Upload each part separately**. Record the time it takes for each upload. Also record the Upload PC specifications. (CPU, memory, etc.)
2. **Query each part separately**, from the same client machine. Record the time it takes for each query. Also record the client PC specifications. (CPU, memory, etc.)
3. **Record these times in a text file**. Name the text file “xx-xx-xx-real-part.txt”, including the date in which the tests were taken.
4. Place this file in the time-test folder, located in the maintenance directory.

6.4 Re-indexing the WERTEVAR table

The WERTEVAR table contains all the individual measured values. This table will grow and change the quickest. The indices on the WERTEVAR table have to be kept up-to-date and functional.

In Oracle versions 7i and 8i, this script should be run on a regular maintenance schedule. How often it should be run depends on the amount of data loaded.

In Oracle version 9i and higher, the WERTEVAR table should be re-indexed after deleting data, or if the indices have corrupted.

Adjust the script to the used tablespace names and indices.

Connect to the database with the table owner (QDAS) and run the script with the command:

```
START <Path>\reindex_wertevar.sql;
```

Rebuilding the indices or re-creating the indices can also be done using other Oracle tools.

After re-indexing the WERTEVAR table, the database should be analyzed (analyze and compute statistics).

6.5 Analyzing the Database

To analyze the database and compute statistics, it is recommended that you make the time tests with reference data (see above) and document the results.

Run the analyzer script when **connected as a dba** to the database.
Run this script with the command:

```
START <Path>\analyze_qdastables.sql;
```

Note: Depending on the SQL Plus client used (16 or 32-bit), the size of the file name may need to be altered.

After the analyzer has run, it is recommended that you make the time tests again and compare and document the results. These results should be documented in the same files that the previous time tests were taken. When documenting, make note of the time tests before and after running the analyzer script.

This script only refers to the biggest QDAS tables. Analyzing more tables or the entire database using Oracle tools is possible also. It should be run on a regular maintenance schedule. How often it should be run depends on the amount of data loaded.

7 History of Changes

The history of changes is customer specific. This chapter should include every change to the database configuration, hardware configuration or software configuration.
It is **highly recommended** to document the changes for querying and saving specific parts in the database before and after the changes.

Changes on July 30th 2002:

- Check the contents of the database first (**see Information about database contents**)

- Take the index for the measurement values (typical QWERT_INDEX) offline and move it to another logical disk drive. (copying the index file with the operating system , you must be logged in as dba).
  - ⇨ see Script MOVEINDEXTABLESPACE.SQL

- Drop all indexes related to table wertevar and rebuild the indexes
  - ⇨ see Script REINDEX_WERTEVAR.SQL

- Run the analyzer
  - ⇨ see Script ANALYZE_QDASTABLES.SQL

- Copy the new autodbup.exe over to the old version
  Date/Time of the new AUTODBUP.EXE = ????/????
  - ⇨ (Changes are made for opening the table WERTEVAR optimized, because the index was not used properly).

8 Appendix

8.1 Example of DBINFO.sql data

This is a partial example of the data produced from the DBINFO.sql script.

```
Sever Name:

Database Name (SID): BLOCKDB

Version/SGA Information:

Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
PL/SQL Release 8.1.7.0.0 - Production
CORE   8.1.7.0.0          Production
TNS for 32-bit Windows: Version 8.1.7.0.0 - Production
NLSRTL Version 3.4.1.0.0 - Production

Total System Global Area as of July 16th, 2002:    185,432,092 bytes

Tablespaces and Data files:

         File                                  Size
Tablespace  ID              Datafile              in MG.
---------- ---- -------------------------------------------- -----------
INDX        6 E:\ORA81\ORADATA\BLOCKDB\INDX01.DBF              58.00
**********                                     -----------
TS SIZE:                                            58.00

QDAS_DEFAU    7 C:\BLOCKDB\QDAS_DEFAULT.DBF                      3.00
**********                                     -----------
TS SIZE:                                             3.00

QMERK_INDE   11 C:\BLOCKDB\QMERK_IND.DBF                     10.00
**********                                     -----------
TS SIZE:                                            10.00

QMERK_TABL    9 C:\BLOCKDB\QMERK_TAB.DBF                     30.00
**********                                     -----------
TS SIZE:                                            30.00

QWERT_INDE   10 E:\ORA81\ORADATA\BLOCKDB\QWERT_IND.DBF          2,130.00
**********                                     -----------
TS SIZE:                                          2,130.00

QWERT_TABL    8 C:\BLOCKDB\QWERT_TAB.DBF                     380.00
**********                                     -----------
TS SIZE:                                           380.00

RBS         2 E:\ORA81\ORADATA\BLOCKDB\RBS01.DBF             520.00
**********                                     -----------
TS SIZE:                                           520.00
```

Etc.

8.2 Example of the part_char_valuesinfo.sql data

This is a partial example of the data produced from the part_char_valuesinfo.sql script.

```
PARTKEY PARTDESC           PARTNO        CHAR_COUNT
--------- ------------------- --------------- ----------
      50 L6 BLK INL 140A      12563712           7
      56 L6 BLK INL 140B      12563712           7
      51 L6 BLK INL 140C      12563712           7
      54 L6 BLK INL 150A      12563712           6
      52 L6 BLK INL 150B      12563712           6
      44 L6 BLK INL 170A_AUD  12563712              36
      45 L6 BLK INL 170A_PROD 12563712               11
      46 L6 BLK INL 170B_AUD  12563712              36
      48 L6 BLK INL 170B_PROD 12563712               11
      80 L6 BLOCK           12563711           8
      77 L6 BLOCK OP 10       12563712           64
      68 L6 BLOCK OP 120      12563712          144
      69 L6 BLOCK OP 130      12563712          158
      70 L6 BLOCK OP 140      12563712          176
```

```
  PARTKEY PARTDESC           PARTNO        VALUE_COUNT
--------- ------------------- --------------- -----------
      50 L6 BLK INL 140A      12563712         400785
      56 L6 BLK INL 140B      12563712         374535
      51 L6 BLK INL 140C      12563712         469238
      54 L6 BLK INL 150A      12563712         546276
      52 L6 BLK INL 150B      12563712         563022
      44 L6 BLK INL 170A_AUD  12563712             395568
      45 L6 BLK INL 170A_PROD 12563712              913715
      46 L6 BLK INL 170B_AUD  12563712             287136
      48 L6 BLK INL 170B_PROD 12563712              905509
      80 L6 BLOCK           12563711           20
      77 L6 BLOCK OP 10       12563712           4352
      68 L6 BLOCK OP 120      12563712          20298
      69 L6 BLOCK OP 130      12563712          26228
      70 L6 BLOCK OP 140      12563712          27383
```

Etc.

8.3 Example of Hardware Configuration Documentation

This is to show an example of recording the hardware that is being time tested.

PC Name	Make/Model	CPU	Memory	OS	Function
Server	NEC	P3 450 mhz	512	Win2K Server	Q-DAS Oracle Server All Instances
PC1	Gateway E-3200	P2 350 mhz	256	Win98 2nd edition	Viewer PC 16 + 32-bit qs-STAT All Instances
PC2	Gateway E-3200	P2 350 mhz	228	Win2K Pro sp2	Viewer PC 32-bit qs-STAT crankdb

Etc.

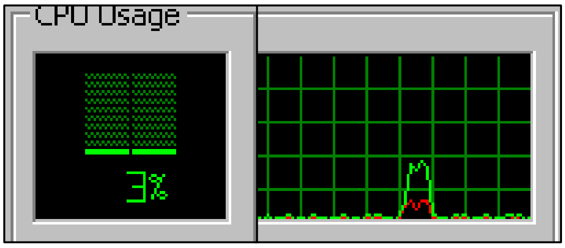
9 Knowledge Base

9.1 qs-STAT Millennium

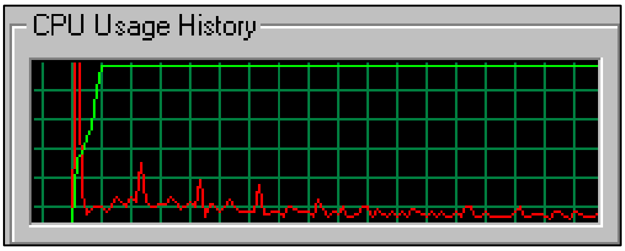
1. Symptom:
While querying the database, it was noticed that the server CPU was idle and the client CPU was at 100%. The only time the server CPU showed activity was when it changed characteristics (there were 7 characteristics in the part and 7 peaks in the server CPU history) and the data was transferred to the client (verified on the network switch). (See Server-A)

This also seemed to happen when the queried part contained few characteristics but with large amounts of values in each characteristic.

Server-A



Client-A

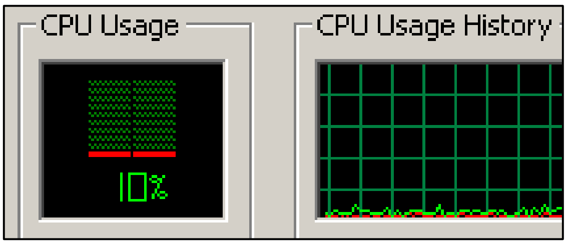


Solution:
A change was made in the source code for qs-STAT Millennium, regarding the server-cursor and how the server handles the query.

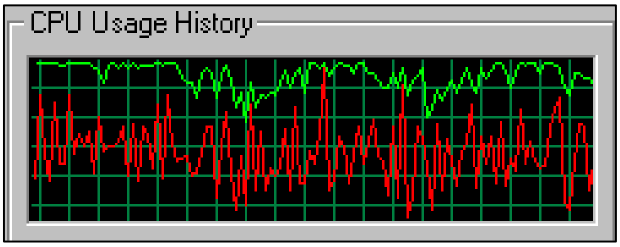
After this change, it was noticed that the server CPU was more active during the entire query time and there were no significant peaks in processor usage. (See Server-B)
Also, data was being sent over the network continuously to the client. (verified on the network switch)

The client CPU time was not at 100% during the entire query. (See Client-B)

Server-B



Client-B



There was an improvement in overall query time with the server-cursor change.

2. Symptom
Adding new fields to the database
Solution:
When you add fields to the database manually, you need to run the wv_insert script. The new tables script includes the contents of this script, but you need to run wv_insert script by itself after adding fields to the database.

3. Symptom
When setting up a new database, the query speed when loading a part was very low.
Solution:
The sort area size of the database memory configuration was changed from 64 KB to 16 MB. This was changed directly in the init_db.ora file for the database instance.

9.2 32-Bit Configuration (.INI) File (qsstat2000.ini)

Speed up query time:

Change *swap_breite_urwerte=16* to *swap_breite_urwerte=512 (or to a higher value depending on the estimated average number of values per characteristic to be loaded)*
If this entry is not present in the file, add it.

NOTE: This setting speeds up the query time, but slows down the file loading time. This should **only be used for a client PC, NOT an upload PC**.

9.3 Q-DAS Database Values Index Move

It was recommended to split the physical location of the database files and the database index files. It was decided and proven successful that moving the largest and busiest index, *wertevar.ind*, was needed.

RAID-5
Moving the location of the Index file was tested on a RAID-5 build PC. There were three OS partitions (C:\, D:\, E:\) created across the RAID-5 build. The database data files were located on the E:\ partition and the index file was moved to the D:\ partition.
We did not see a dramatic increase in performance when we first moved the index to a different partition than the database data files are located.
Analyzer
After moving the index files, we ran an Oracle command "*analyze*" on the index file. After we ran the analyze command, the performance was greatly improved.

2 Physical Hard drives
On a different PC with two physical hard drives, we moved the index file to another hard drive. (the database data files and index files were located on two separate hard drives). This immediately improved performance. The index files were then dropped and the tables were re-indexed. This increased performance even more.

9.4 Imported Database Performance

An import of a database was performed. After the import, an upload test was performed. The upload speed was fast. The database server was turned off and rebooted. After the restart, the upload speed was very slow. The analyzer script was run and the upload speed was fast again.

Testing Documentation
July 22-31, 2002

Hardware Configurations

PC Name	Make/Model	CPU	Memory	OS	Function	IP Address 10.10.10.x
showserver	NEC	P3 450 mhz	512	Win2K Server	Q-DAS Oracle Server All Instances	.4
PC1	Gateway E-3200	P2 350 mhz	256	Win98 2nd edition	Viewer PC 16 + 32-bit qs-STAT All Instances	.3
PC2	Gateway E-3200	P2 350 mhz	228	Win2K Pro sp2	Viewer PC 32-bit qs-STAT cranckdb	.2
blackpc	DataMyte	AMD-K6 300 mhz	160	Win98 2nd edition	Viewer PC 16 + 32-bit qs-STAT All Instances 16-bit Upload All Instances	.1
Michael's laptop		P3- 1 Ghz	256	Win2K Server	Viewer PC All Instances	.15
PC4	Gateway E-3200	P2 350 mhz	327	Win NT	Viewer PC and Upload 16+32-bit All Instances	.6
	IBM Netfinity 5600 RAID-5	Dual 500 mhz	256	Win NT Server	FES Oracle Server Block and Head Instances	.10

Query Times - Compare Q-DAS Inc. Server and Value Partitioning

PARTKEY	PARTDESC	PARTNO	CHAR_COUNT	VALUE_COUNT	Q-DAS PC4 32bit	Q-DAS PC4 32bit		Michael 32bit	Q-DAS PC1 16bit	
					breite_urwerte=16	breite_urwerte=512				
					QUERY TIME IN SECONDS					
64	01/23/02-06/03/02	M140C	6	2112	3			2	5	
55	02/08/02-06/02/02	M090	33	55836	112	60	70	23	86	
74	02/14/02-06/04/02	M090	33	6633	8					
60	02/17/02-06/01/02	M160	507	20280	50	50	50	19	110	
63	03/25/02-05/09/02	M140A	6	768	2					
28	04/10/01-06/04/02	M120	2	7458	10					
79	04/19/02-06/03/02	M140A	6	282						
31	05/01/01-02/27/02	M140A	6	1854						
32	05/12/01-01/11/02	M140C	6	2418						
30	05/19/01-06/03/02	M140B	6	2652						
53	06/19/01-03/20/02	M140A	6	426						
26	07/31/00-06/04/02	M010	20	53220	90	55	66	15	75	
41	08/28/01-01/29/02	M160	507	33462	68	51	55	20	115	
47	09/11/01-06/04/02	M020	4	9140						
29	11/09/00-05/23/02	M130	8	9896						
34	11/29/00-01/08/02	M090	33	152856	390	205	216	90	218	
50	L6 BLK INL 140A	12563712	7	400785	2580	2600		900	570	
56	L6 BLK INL 140B	12563712	7	374535						
51	L6 BLK INL 140C	12563712	7	469238						
54	L6 BLK INL 150A	12563712	6	546276						
52	L6 BLK INL 150B	12563712	6	563022						
44	L6 BLK INL 170A_AUD	12563712	36	395568						
45	L6 BLK INL 170A_PROD	12563712	11	913715						
46	L6 BLK INL 170B_AUD	12563712	36	287136						
48	L6 BLK INL 170B_PROD	12563712	11	905509						
80	L6 BLOCK	12563711	8	20						
77	L6 BLOCK OP 10	12563712	64	4352						
68	L6 BLOCK OP 120	12563712	144	20298						
69	L6 BLOCK OP 130	12563712	158	26228						
70	L6 BLOCK OP 140	12563712	176	27383						
71	L6 BLOCK OP 150	12563712	129	42024	60	32	53	15	75	
72	L6 BLOCK OP 160	12563712	80	1054						
73	L6 BLOCK OP 170	12563712	144	1110						
75	L6 BLOCK OP 20	12563712	270	17143						
78	L6 BLOCK OP 30	12563712	66	4950						
81	L6 BLOCK OP 40	12563712	202	1818						
67	L6 BLOCK OP 50	12563712	194	28324						
57	L6 BLOCK OP 90	12563712	720	99007	180	108	140	60	236	
87	L6 BLOCK OP010M	12563711	4	120						
85	L6 BLOCK OP020M	12563711	4	444						
86	L6 BLOCK OP090M	12563710	33	990						
88	L6 BLOCK OP120M	12563710	2	96						
89	L6 BLOCK OP130M	12563710	8	96						
82	L6 BLOCK OP140AM	12563710	6	90						
83	L6 BLOCK OP140BM	12563710	6	162						
84	L6 BLOCK OP140CM	12563710	6	126						

Query Times- Compare FES Server and Index on different Partition

PARTKEY	PARTDESC	PARTNO	CHAR_COUNT	VALUE_COUNT	Q-DAS PC4 32bit breite_urwerte=16	Q-DAS PC4 32bit breite_urwerte=512		Michael 32bit	Q-DAS PC1 16bit
					QUERY TIME IN SECONDS				
64	01/23/02-06/03/02	M140C							
55	02/08/02-06/02/02	M090				57	78		94
74	02/14/02-06/04/02	M090							
60	02/17/02-06/01/02	M160				55	49		108
63	03/25/02-05/09/02	M140A							
28	04/10/01-06/04/02	M120							
79	04/19/02-06/03/02	M140A							
31	05/01/01-02/27/02	M140A							
32	05/12/01-01/11/02	M140C							
30	05/19/01-06/03/02	M140B							
53	06/19/01-03/20/02	M140A							
26	07/31/00-06/04/02	M010				68	66		85
41	08/28/01-01/29/02	M160				56	55		116
47	09/11/01-06/04/02	M020							
29	11/09/00-05/23/02	M130							
34	11/29/00-01/08/02	M090				222	223		230
50	L6 BLK INL 140A	12563712							592
56	L6 BLK INL 140B	12563712							
51	L6 BLK INL 140C	12563712							
54	L6 BLK INL 150A	12563712							
52	L6 BLK INL 150B	12563712							
44	L6 BLK INL 170A_AUD	12563712							
45	L6 BLK INL 170A_PROD	12563712							
46	L6 BLK INL 170B_AUD	12563712							
48	L6 BLK INL 170B_PROD	12563712							
80	L6 BLOCK	12563711							
77	L6 BLOCK OP 10	12563712							
68	L6 BLOCK OP 120	12563712							
69	L6 BLOCK OP 130	12563712							
70	L6 BLOCK OP 140	12563712							
71	L6 BLOCK OP 150	12563712				43	52		83
72	L6 BLOCK OP 160	12563712							
73	L6 BLOCK OP 170	12563712							
75	L6 BLOCK OP 20	12563712							
78	L6 BLOCK OP 30	12563712							
81	L6 BLOCK OP 40	12563712							
67	L6 BLOCK OP 50	12563712							
57	L6 BLOCK OP 90	12563712				126	128		255
87	L6 BLOCK OP010M	12563711							
85	L6 BLOCK OP020M	12563711							
86	L6 BLOCK OP090M	12563710							
88	L6 BLOCK OP120M	12563710							
89	L6 BLOCK OP130M	12563710							
82	L6 BLOCK OP140AM	12563710							
83	L6 BLOCK OP140BM	12563710							
84	L6 BLOCK OP140CM	12563710							

Q-DAS Server 16-bit Upload Test
with index on different HD

Full day's production from GMPT FES

PC	Department	Gauge	Upload Time (Mins)
Black PC	Block	Zeiss	48
PC1	Block	Zeiss	37
Black PC	Block	Marposs	24
PC1	Block	Marposs	23
Black PC	Head	Zeiss	88
PC1	Head	Zeiss	71
Black PC	Head	Marposs	8
PC1	Head	Marposs	6

FES Server 16-bit Upload Test

Full day's production from GMPT FES

PC	Department	Gauge	Upload Time (Mins)
Black PC	Block	Zeiss	Very Long Time
PC1	Block	Zeiss	
Black PC	Block	Marposs	40
PC1	Block	Marposs	
Black PC	Head		
PC1	Head		
Black PC	Head		
PC1	Head		

Q-DAS Server 16-bit Upload Test

with index on different HD and Value Part.

Full day's production from GMPT FES

PC	Department	Gauge	Upload Time (Mins)	After Analyzer	Changed Temp Tablespace
Black PC	Block	Zeiss	25 minutes for 4 files Stopped Upload		
PC1	Block	Zeiss		38	38
Black PC	Block	Marposs			
PC1	Block	Marposs		25	

FES Server 16-bit Upload Test

with Index on Different Partition

Full day's production from GMPT FES

PC	Department	Gauge	Upload Time (Mins)	Index back to RAID partition (from independent drive) and ran Analyzer	New Source Change- Includes all PK's for better Index Search	Index to Physical HD	32-bit Upload
Black PC	Block	Zeiss	47		30	30	
PC1	Block	Zeiss					
Black PC	Block	Marposs	54	158	13	13	
PC1	Block	Marposs					
PC4	Block	Zeiss					12
PC4	Block	Marposs					10

10 Document Control

Version	Edits/Additions	Content	Author
8/9/2002 V: EB	Add.-Section 9	Document Control	J. Kohl
	Add.-Section 5.4	Note: SQL Plus Client and File Size	J. Kohl
	Edit-Section 1.2.1.2	"Start <Path>\init_tablespaces.sql"	J. Kohl
	Add.-Section 8	32-bit INI file Imported Database performance	J. Kohl
10/1/2002 V: EC	Edit-Section 1.2.1.1	Edited the specified contents of the Init_db.sql script	J. Kohl
	Add.-Knowledgebase	16-bit AutoUpload wv_insert script after adding fields	J. Kohl
11/20/2002 V: ED	Add- Script modifications Knowledgebase Other material	Updated the Script content, entire document	M. Wagner
12/10/02 V: EE	Edit- Section 2.2 Decision Tree	Edited the Decision tree to include "PartCharacteristic_NewField.sql"	J. Kohl
	Add.-section 8.3 Knowledgebase	16-bit Upload Number of folders in Upload directory	J. Kohl
	Edit- section 8.2	Grammar and details	J. Kohl
09/06/2002 V: EF	Edit - section 1.1	Unicode character set	M. Oswald
	Edit - section 2.2	Decision tree including update from earlier qs-STAT millennium versions	M. Oswald
	Edit - section 2.2.1	Field name added to "no-delete-columns"	M. Oswald
1/15/2007 V: EG	Edit – section 1.1	New screen shots from 9i, added remark about character set for Asian languages	E. Fassbeck
9/13/2008 V: EH	Edit – section 1.2	Re-arranged instructions, added new script names	E. Fassbeck
	Edit – section 2	Re-arranged instructions	E. Fassbeck
	Added section 2.3	Updating from old ME to newer ME version	E. Fassbeck
	Added section 5.4	Re-index WERTEVAR	E. Fassbeck