



Benutzer Handbuch

DELMIA Process Engineer®

ORACLE 10g R2 Installation



Inhaltsverzeichnis

ORACLE 10g R2 Installation	1
Inhaltsverzeichnis	2
Einleitung	4
Wie setzen Sie nun dieses Handbuch ein?	4
Wie sollte dieses Handbuch gelesen werden?	5
Wie Sie Zeichen und Symbole lesen	6
Vorraussetzungen für die Installation	7
Meine erste Datenbank	8
Migrieren einer ORACLE 9i Datenbank	9
Deinstallation von ORACLE 9i auf Windows	10
Stoppen der laufenden Dienste	10
Entfernen der ORACLE Einträge aus der Registrierungsdatenbank	10
Löschen der ORACLE Service Einträge aus dem CURENT_CONTROL_SET	11
Entfernen der ORACLE Dateien	11
Entfernen der ODBC Datenquellen	11
Installieren der ORACLE Software	12
Client Einstellungen	12
Anlegen einer Datenbank	13
Speicherbereiche	13
Undo Management	14
Neue Tablespaces	15
Schema	17
Log Parameter	18
SPFILE	18
Instanzeinstellungen	19
Informationen zu einigen ORACLE Parametern	19
Indizes	24
Export und Import	25
Exportieren von Daten aus der Datenbank	25
Importieren von Daten in die Datenbank	26
Import der Process Engineer Datenbank	27
Benutzerschema	27
Berechtigungen	28
Tablespaces für Benutzer	29
Rollen anlegen	29

Blob Tabellen erzeugen	30
Data-Pump	31
Hinweise zur Bedienung	31
Export 31	
Import 32	
Process Engineer Datenbank	33
Verwendung des Datenbankassistenten von DPE	36
Hinweise zur Wartung der Datenbank	37
Performance-Tuning	37
Optimierung der Datenbank-Instanz	37
Wartungsarbeiten	38
Einfache Überwachungstätigkeiten	38
Anhang	40
Ergänzend zum Kapitel ‚Blob Tabellen erzeugen‘	40

Einleitung

Die Bedienung, Funktionsweise und Menüführung, die in diesem *Handbuch für die ORACLE Installation* beschrieben ist, wird Ihnen in diesem Handbuch auf einfache und verständliche Weise erklärt. Es zeigt auf, wie Sie Ihre ORACLE Datenbank installieren und einrichten, um darauf mit DELMIA Software zu arbeiten.

Wie setzen Sie nun dieses Handbuch ein?

Dieses Handbuch ist an drei Benutzergruppen gerichtet und diese sollten jeweils nur gezielt spezielle Kapitel auswählen. Die erste Benutzergruppe sind diejenigen, die keine Vorkenntnisse über Datenbanken mitbringen und ORACLE auf Windows installieren wollen. Diese sollten die folgenden Kapitel lesen:

- Voraussetzungen für die Installation
- Meine erste Datenbank
- Process Engineer Datenbank

Diese Kapitel versetzen Sie in die Lage in wenigen Minuten eine typische Datenbank zum Betrieb der DELMIA Software zu installieren und zu konfigurieren. Es sind dazu keine besonderen Vorkenntnisse notwendig. Der Vorteil ist, dass Sie in kurzer Zeit anfangen können mit der DELMIA Software zu arbeiten.

Hinweis: Allerdings werden hierzu Standard-Parameterwerte verwendet, die den durchschnittlichen Datenbankgrößen genügen. In jedem Einzelfall sollte jedoch ein Oracle Datenbankadministrator prüfen, welche Anforderungen genau in seinem Fall vorliegen und ggf. diese Werte anpassen. Nähere Informationen hierzu finden Sie in der offiziellen Dokumentation von ORACLE.

Die zweite Benutzergruppe sind diejenigen, die Experten auf dem Gebiet ORACLE sind oder ORACLE unter UNIX einsetzen wollen. Diese sollten folgende Kapitel lesen:

- Voraussetzungen für die Installation
- Installieren der ORACLE Software
- Anlegen einer Datenbank
- Importieren der Process Engineer Datenbank

Die dritte Benutzergruppe sind diejenigen, die Experten auf dem Gebiet ORACLE sind und die bereits eine DELMIA Datenbank installiert haben und lediglich eine Migration auf V5R17 durchführen. Diese sollten folgende Kapitel lesen:

- Migrieren einer ORACLE 9i Datenbank
- Upgrade einer vorhandenen Datenbank
- Export und Import

Hinweis: Lesen Sie diese Einleitung besonders gründlich durch. Lassen Sie sich führen: Verwenden Sie dazu das Inhaltsverzeichnis, die Überschriften und die Kopfzeile und folgen auch den Querverweisen, die Ihnen weitere Informationen liefern.

Wie sollte dieses Handbuch gelesen werden?



Dieses Benutzerhandbuch richtet sich in erster Linie an Systemadministratoren oder gleichberechtigten Mitarbeitern. Die nachfolgend beschriebenen Funktionen müssen vor der Installation des DELMIA Process Engineers bzw. des ENOVIA Manufacturing Hubs durchgeführt werden.



Die Lektüre dieses Handbuch ist notwendig, um den *ENOVIA Manufacturing Hub* erfolgreich auf der ORACLE Datenbank laufen zu lassen. Es gibt jedoch weiterführende Hinweise zu Installation, zur Performance und weiteren Themen im Handbuch , Administration'.

Wie Sie Zeichen und Symbole lesen

Die Zeichen und Symbole, die in diesem und in allen weiteren Handbüchern verwendet werden, dienen nicht nur zur allgemeinen Verschönerung eines Handbuchs, obwohl das auch eine der Aufgaben ist, sie dienen vor allem der Benutzerführung, um Ihnen den Inhalt auf leicht verständliche Weise zu erklären. Kapitel und Kapitelabschnitte werden durch Überschriften eingeleitet. Die Überschriften haben entsprechend der Verwendung unterschiedliche Schriftgrößen.

Nachfolgend wird Ihnen die Bedeutung der Symbole erklärt:



Mit diesem Symbol werden Textstellen bezeichnet, die den Funktionsumfang beschreiben, den Sie in einem Kapitel kennen lernen werden. Es steht daher in der Regel am Anfang eines Kapitels oder Abschnitts. Zudem werden wichtige Textstellen mit diesem Zeichen hervorgehoben.

Hinweis: Mit diesem Symbol werden Hinweise gekennzeichnet, die zu einem Thema noch zusätzliche Informationen liefern, die für das Weiterarbeiten sehr wichtig sind. Das Hinweis-Zeichen kann sowohl an einem Kapitelanfang oder bei einer bestimmten Textstelle im Kapitel stehen. Die Texte, die mit diesem Zeichen eingeleitet werden, sind zusätzlich mit dem Wort *Hinweis* gekennzeichnet. Der Text selbst ist immer kursiv geschrieben.

Achtung: Mit diesem Symbol werden Sie auf Sachverhalte aufmerksam gemacht, die zu möglichen Fehler bei der Bedienung des Programms führen könnten, und die Sie daher beachten sollten. Das Achtung-Zeichen kann sowohl an einem Kapitelanfang oder bei einer bestimmten Textstelle im Kapitel stehen. Die Texte, die mit diesem Zeichen eingeleitet werden, sind zusätzlich mit dem Wort **Achtung** gekennzeichnet. Der Text selbst ist immer kursiv geschrieben.

Beispiel

Mit diesem Symbol werden Sie auf Beispiele aufmerksam gemacht, die einen Sachverhalt verdeutlichen

- ➡ Mit diesem Symbol werden die einzelnen Bedienschritte einer Handlungsanweisung gekennzeichnet. Mit Handlungsanweisungen werden Bedienschritte beschrieben, um beispielsweise ein Menü zu öffnen oder eine Funktion auszuführen.
- Mit diesem Symbol werden Aufzählungen gekennzeichnet. Das Aufzählungssymbol kann sowohl für eine Gliederung eines Fließtextes verwendet werden sowie stichpunktartig Themenschwerpunkte aufzulisten.



Mit diesem Symbol werden Sie darauf aufmerksam gemacht, dass es zu diesem Thema noch weitere Informationen in einem anderen Handbuch gibt.

Voraussetzungen für die Installation

Sie sollten folgende Voraussetzungen geklärt haben, bevor Sie mit der Installation beginnen :

Installationsvoraussetzung

- DVD mit ORACLE 10g Database Release 2 (empfohlen für die unterstützten Windows Versionen)
- ORACLE 10g Database Release 2 Enterprise Edition or Standard Edition

Hinweis: ORACLE Version 10.2.0.2 als Minimalanforderung für R18

Hardwarevoraussetzung

- Mindestens 2400 MHz CPU, Mindestens 2 GByte Hauptspeicher
- 10 GB freier Speicherplatz auf der Festplatte

Softwarevoraussetzung

- Windows XP, Windows 2003 Server oder Unix

Meine erste Datenbank

- Bitte legen Sie die **DVD** mit ORACLE 10g Database Release 2 Standard Edition in Ihr CD Laufwerk der Maschine, auf der Sie die Datenbank installieren müssen.
- Starten Sie das Programm **setup.exe** im Hauptverzeichnis der DVD und folgen Sie den Installationsanweisungen am Bildschirm.

Migrieren einer ORACLE 9i Datenbank

Seit der Version PE R17 wird **ORACLE 9i und ORACLE 10g** unterstützt. Dazu müssen Sie die Datenbank auf ORACLE 10g migrieren.

⇒ Zudem müssen Sie auf jedem Oracle Client (auf allen Rechnern auf denen der PPRServer installiert wurde) ORACLE 10g migrieren.

Mit der ORACLE 9i kam der Aspekt des „Locally Managed Tablespaces hinzu, insbesondere des „System Tablespaces“.

Allerdings erreicht man einen Locally Managed SYSTEM - Tablespace nicht durch Migration einer Datenbank, sondern ausschließlich durch Neuerstellung mit ORACLE 9i Release2 oder Oracle 10g Release2.

Eine Migration ist ein sehr komplexer und oft auch zeitintensiver Vorgang. Es ist notwendig vor einer Migration die aktuellen Readme Dateien des Oracle Supports lesen. Ebenso möchten wir an dieser Stelle auf einschlägige Oracle Literatur verweisen.

Nachfolgend die unterstützen Versionen, die den Mindestanforderungen entsprechen, und von denen direkt auf ORACLE 10g R2 gewechselt werden kann:

- 9.2.0.4/9.2.0.5/9.2.0.6/9.2.0.7

Hinweis: Empfohlen wird eine Migration mittels der herkömmlichem Export/Import Funktionalität. Obwohl mehr Speicherplatz notwendig ist und die Migration möglicherweise länger dauert, ergeben sich folgende Vorteile:

- Reorganisation der Objekte wird durchgeführt
- Falls der Exportierte Dump nicht explizit gelöscht wurde kann bei Problemen auf die alte Version zurückgegangen werden

Deinstallation von ORACLE 9i auf Windows

Sie können die ORACLE 9i auf der Datenbank und auf den DELMIA PPR Maschinen deinstallieren. Wenn Sie jedoch keine Ressourcenprobleme bezüglich des Plattenplatzes oder der CPU haben, können Sie auch ORACLE 9i installiert lassen.

Hinweis: Bevor Sie ORACLE deinstallieren, sollten Sie ein Backup Ihrer Datenbank vornehmen. Dies ist Aufgabe eines ORACLE Administrators. Für ein Backup ihrer Daten lesen Sie bitte das Kapitel [Export und Import](#).

Stoppen der laufenden Dienste

- Starten Sie den Dialog zur Anzeige der Dienste auf Ihrem Betriebssystem.
- *Start->Verwaltung->Dienste*
- Suchen Sie in diesem Dialog nach Diensten mit dem Prefix ORACLE, die den Status 'Started' haben und stoppen Sie diese, indem Sie im Kontextmenü jeweils 'Stop' auswählen.

Entfernen der ORACLE Einträge aus der Registrierungsdatenbank

Die von ORACLE eingefügten Einträge in der Windows Registrierungsdatenbank finden Sie unter dem Schlüssel:

⇒ HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\Homex

Eine durch DELMIA früher installierte Oracle 9i Datenbankinstanz können Sie anhand des Eintrages ORACLE_SID erkennen, falls dieser den Wert 'O9IE' enthält.

- ⇒ Notieren Sie sich in diesem Fall den Pfad der unter dem Schlüssel ORACLE_HOME eingetragen ist. Dies ist das oberste Verzeichnis unter dem sich die zu löschenden Dateien der zu ersetzenden ORACLE Installation befinden.
- Löschen Sie nun folgenden Schlüssel:
- ⇒ HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\Homex

Löschen der ORACLE Service Einträge aus dem CURENT_CONTROL_SET

- ➊ Löschen Sie die Schlüssel, die im Wert ImagePath, den Pfad Ihrer zu löschenden Instanz eingetragen haben. Die Schlüssel heißen:
- ⇒ HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ORACLE...

Entfernen der ORACLE Dateien

Im Dateieexplorer löschen Sie den Inhalt und alle Unterverzeichnisse im Verzeichnis ORACLE_HOME, dessen Wert Sie sich im Schritt ‚Entfernen der Einträge der Registrierungsdatenbank‘ notiert haben.

Entfernen der ODBC Datenquellen

- ➊ In der Registrierungsdatenbank suchen Sie den Schlüssel
 - HKEY_LOCAL_MACHINE->Software->ODBC->ODBCINST.INI
- ➋ Suchen Sie hier nach einem Treiber mit dem Prefix ORACLE.
 - ⇒ Notieren Sie davon den aktuellen Wert des Feldes Driver (z.B. D:\ORACLE\ORA92E\BIN\SQORA32.DLL).
- ➌ Wiederholen Sie das für die Schlüssel mit dem Präfix ORACLE.

Löschen Sie danach alle vorhandenen Schlüssel mit dem Präfix ORACLE.
Löschen Sie ebenfalls Zeichenketten unter dem Schlüssel,

 - HKEY_LOCAL_MACHINE->Software->ODBC->ODBCINST.INI->ODBC Drivers

die auf Schlüssel mit dem Präfix ORACLE verweisen.
- ➍ In der Registrierungsdatenbank suchen Sie den Schlüssel
 - HKEY_LOCAL_MACHINE->Software->ODBC->ODBC.INI
- ➎ Löschen Sie Schlüssel, die im Feld Driver den oben notierten Wert haben und damit auf einen ORACLE-Treiber verweisen.
- ➏ In der Registrierungsdatenbank suchen Sie den Schlüssel
 - HKEY_LOCAL_MACHINE->Software->ODBC->ODBC.INI->ODBC Data Sources
 - ⇒ Löschen Sie wiederum alle Zeichenfolgen, deren Wert auf die von Ihnen gelöschten ORACLE-Schlüssel verweisen.

Installieren der ORACLE Software

Hinweis: ORACLE-Einstellungen sollten Sie vor der Installation der DELMIA-Software und der Einrichtung des PPR-Hubs mit Ihrem Administrator klären und besprechen.

Die in diesem Handbuch festgehaltenen Werte dienen lediglich dazu, bereits bekannte und bewährte Einstellungen allen Kunden zugänglich zu machen. Einstellungen und Vorgaben für den korrekten Betrieb der DELMIA-Software sind *kursiv* und farblich hervorgehoben.

Client Einstellungen

Auf jeder PPR Server Maschine läuft ein ORACLE Client. Stellen Sie für jeden ORACLE Client die Zeit für das Beenden der Sessions auf fünf Minuten ein.

Dazu muss in der Datei <ORACLE-Pfad>\network\admin\sqlnet.ora folgender Wert eingetragen sein:

```
SQLNET.EXPIRE_TIME=5
```

Für jeden ORACLE Client wird ein Dedicated Server Prozess gestartet

Zu einem Dedicated Server System gehören:

tnsnames.ora: Konfigurationsdatei zur Umsetzung des Net Service Namens in eine Server Adresse

sqlnet.ora: Konfigurationsdatei auf Client- und Serverseite (optional)

Listener: Serverseitiger Prozess, der auf Verbindungswünsche reagiert.

Anlegen einer Datenbank

Speicherbereiche

Locally Managed Tablespaces und Automatic Segment-Space Management (ASSM)

- Locally Managed Tablespaces wurden mit ORACLE 8i eingeführt.
- ASSM mit ORACLE 9i.
- Nur permanente Locally Managed Tablespaces können mit ASSM erzeugt werden.
- Automatic Segment Space Management legt fest wie ein freier und belegter Speicherplatz verwaltet werden kann.
- Die Extent Verwaltung über Locally Managed Tablespaces mit AUTOALLOCATE Option ist der Versuch seitens ORACLE alle Storage-Parameter überflüssig zu machen.

Zur Wahl stehen die Optionen:

- MANUAL
- AUTO

Um genügend Informationen über *Locally Managed Tablespaces* und ASSM zu erhalten ist, das Wissen eines **zertifizierten** ORACLE Datenbank Administrators notwendig.

Hinweis: An dieser Stelle wollen wir darauf hinweisen, dass es bei der Arbeit mit *Locally Managed Tablespaces* und Automatic Segment Space Management im Zusammenhang mit Lobs (Large Object Blocks) möglicherweise Fehler auftreten können. Diese sind innerhalb der ORACLE Fehlerdatenbank näher beschrieben. Die entsprechenden Fehlernummern sind 2784201, 3029292, 3213101.

- Lobs werden zum Beispiel bei
- Detailings (Daten die nur im V5-Release zu sehen sind),
- Datenkarten und Skripten verwendet.

Undo Management

Für die Arbeit mit dem Undo Management werden Tablespaces benötigt. Die Kennzeichnung der Tablespaces mit Namen können individuell vom Datenbank-Administrator (DBA) beim jeweiligen Kunden gewählt werden.

- Für die *Version 9* empfiehlt es sich, einen Undo-Tablespace anzulegen und in der Initialisierungsdatei den Parameter `UNDO_MANAGEMENT=AUTO` zu setzen.
- Um zwischen den Modi umzuschalten ist ein Neustart der Datenbank notwendig.
- Es ist nicht möglich beide Methoden innerhalb einer Datenbankinstanz zu verwenden. Wurde das automatische Undo Management aktiviert, werden die Rollback-Segmente vom System verwaltet und das explizite Anlegen von Rollbacksegmenten entfällt.
- Als Undo-Tablespace wird der erste in der Datenbank gefundene Tablespace vom Typ UNDO genutzt. Optional kann durch den Parameter `UNDO_TABLESPACE` in der Initialisierungsdatei ein bestimmter Undo-Tablespace gesetzt werden.

Folgende ORACLE Parameter werden nicht mehr verwendet, wenn Sie das automatische Undo Management verwenden:

- `ROLLBACK_SEGMENTS`
- `TRANSACTIONS`
- `TRANSACTIONS_PER:ROLLBACK_SEGMENT`
- `MAX_ROLLBACK_SEGMENTS`

⇒ Neue ORACLE Parameter, wenn ein Undo Tablespace verwendet wird

UNDO_MANAGEMENT

Wird dieser Parameter auf `AUTO` gesetzt, dann wird der Undo Management Modus verwendet. Falls er auf `MANUAL` gesetzt wird, wird der manuelle Undo Management Modus verwendet (explizit angelegte Rollbacksegmente)

UNDO_TABLESPACE

Das ist ein dynamischer Parameter der den Namen des verwendeten Undo Tablespaces angibt.

UNDO_RETENTION

Gibt die Zeit in Sekunden an, in der Informationen für bereits abgeschlossene Transaktionen aufbewahrt werden. Hiermit wird die Lesekonsistenz für lang laufende Transaktionen sichergestellt.

UNDO_SUPPRESS_ERRORS

Es besteht die Möglichkeit Fehlermeldungen, die mit SQL Statements im manuellen Undo Management ausgegeben werden, für das automatische Undo Management auszublenden. (`TRUE` = ausblenden, `FALSE` = einblenden)

Neue Tablespaces

SYSAUX Tablespace:

Mit ORACLE 10g wurde neben dem SYSTEM Tablespace ein neuer Pflicht-Tablespace, als Ergänzung zum SYSTEM Tablespace eingeführt. Er kann weder umbenannt noch gelöscht werden.

Durch diesen Tablespace sollen:

- System- und Nichtsystem Daten getrennt werden
- der System- Tablespace entlastet werden
- für Komponenten wie das Enterprise Manager Repository, den LogMiner etc. als Speicherort dienen

Der SYSAUX Tablespace ist

- Permanent
- Read Write

Und es gelten folgende Speicherbereiche:

- Extent Management Local
- Segment Space Management: Auto

DEFAULT Table space:

Mit ORACLE 10g kann einer Datenbank nicht nur ein Default Temporary Tablespace zugewiesen werden, sondern auch ein Default Permanent Tablespace. Bisher stand dieser Default grundsätzlich auf SYSTEM.

Die Befehle CREATE DATABASE und ALTER DATABASE wurden entsprechend erweitert, falls beim ‚create user‘ Statement kein Default Tablespace mit angegeben wird, wird der Default Permanent Tablespace verwendet.

Beispiel:

```
SQL> Create database O10g ... default tablespace ERGO_USR datafile ...
```

Zum Setzen nach dem Erzeugen der Datenbank

```
SQL> Alter database O10G Default Tablespace "ERGO_USR";
```

In der View DATABASE_PROPERTIES ist in der Spalte PROPERTY_NAME ein Eintrag ‚default_permanent_tablespace‘, dem jetzt der Wert ‚ERGO_USR‘ zugewiesen sein muss (Spalte: PROPERTY_VALUE).

Bigfile Tablespace (BFT):

Erwähnt werden sollte der Vollständigkeit halber, dass mit der ORACLE 10g ein neuer Tablespace eingeführt wurde, der die Verwaltung erheblich größerer Datenmengen als früher erlaubt. Der bisher bekannte Typ besteht weiter unter der Bezeichnung **Smallfile Tablespace**. Der Default beim Anlegen von Datenbanken ist der Smallfile Tablespace. Ein Bigfile Tablespace sollte immer dann eingesetzt werden, wenn die Datenbank die bisher möglichen 512 Peta-Byte ausgereizt hat.

Neu sind auch so genannte **Temporary Tablespace Groups** (TTG). Bei paralleler Sortierung über mehrere Partitionen und bei einer Single-SQL auf einem Bigfile Tablespace kann das Arbeiten mit einer TTG sinnvoll sein.

***Hinweis:** Nur in sehr extremen und speziellen Einsatzbereichen werden durch die Neuerungen Steigerungen der Performance festzustellen sein. Dagegen bringen die administrativen Erweiterungen wie beispielsweise der SYSAUX Tablespace einiges an Erleichterung durch eine übersichtliche Strukturierung.*

Schema

Folgende Namen werden für ORACLE10g R2 empfohlen:

- Tablespace für UNDO **UNDOTBS1**
Größe: Minimum 200 MB
Storage parameter:
AUTOEXTEND ON NEXT 10M MAXSIZE 32767M
- Tablespace für SYSTEM **SYSTEM**
Größe: Minimum 300 MB
Storage parameter:
AUTOEXTEND ON NEXT 10M MAXSIZE 32767M
- Tablespace für SYSTEM **SYS_AUX**
Größe: Minimum 110 MB
Storage parameter:
AUTOEXTEND ON NEXT 10M MAXSIZE 32767M EXTENT
MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO
- Tablespace für USER **ERGO_USR**
Größe: Minimum 500 MB
Storage parameter:
AUTOEXTEND ON NEXT 50M MAXSIZE 32767M EXTENT
MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO
- Tablespace für INDEX **ERGO_IND**
Größe: Minimum 500 MB
Storage parameter:
AUTOEXTEND ON NEXT 50M MAXSIZE 32767M EXTENT
MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO
- Tablespace für **TOOLS**
Größe: Minimum 10 MB
Storage parameter:
AUTOEXTEND ON NEXT 10M MAXSIZE 32767M EXTENT
MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO
- Tablespace für **TEMPORARY** (Default temporary tablespace)
Größe: Minimum 500 MB
Storage parameter:
AUTOEXTEND ON NEXT 10M MAXSIZE 32767M EXTENT
MANAGEMENT LOCAL UNIFORM SIZE 10M

Log Parameter

Empfohlen werden:

- Drei Redo Logdateiengruppen mit mindestens 2 Dateien, minimale Größe 30MB
- Es sollten etwas 5-10 ORACLE Logswitches pro Stunde erfolgen. Abhängig davon sollte die Größe der Logdatei und die Anzahl der Gruppen von Logdateien ermittelt werden
- Es sollte überwacht werden, dass keine Wartezeiten für die Archivierungsprozesse benötigt werden

SPFILE

Bitte beachten Sie, dass mit der ORACLE 10g Database Release 2 ein binäres Server Parameter File auf dem Server existiert. Im Gegensatz zum text-basierenden PFILE können dort keine Parameteränderungen mit einem Text-editor vorgenommen werden. Oracle empfiehlt die Verwendung dieser SPFILES. Wird kein SPFILE beim Startup gefunden, dann wird das PFILE (init.ora) verwendet.

Änderungen an Parametern in SPFILES werden mit dem ALTER SYSTEM Befehl durchgeführt und mit ,scope = spfile' ins Spfile geschrieben werden

. Vorteile dieser SPFILES sind beispielsweise:

- Persistente Änderungen mit ALTER SYSTEM (wenn gewünscht)
- Oracle kann selber in die SPFILES reinschreiben (Self – Tuning – Parameters)

In Oracle 10g R2 existiert unter dem Windows Betriebssystem ein PFILE im <ORACLE_HOME>/Database Verzeichnis, welches einen Verweis auf das SPFILE im dbs Verzeichnis hat.

Instanzeinstellungen

Beim Anlegen der Instanz sollten folgende Einstellungen gesetzt werden:

- Der Zeichensatz UTF8 ist sehr wichtig, da er nach dem Anlegen der Instanz nicht geändert werden kann.

```
MAXLOGFILES 30
MAXLOGMEMBERS 3
MAXLOGHISTORY 800
MAXDATAFILES 256
MAXINSTANCES 1
CHARACTER SET UTF8
NATIONAL CHARACTER SET UTF8;
```

Hinweis: Der Default Unicode Zeichensatz AL32UTF8 wird seit der V5R18 unterstützt

Informationen zu einigen ORACLE Parametern

- OPTIMIZER_MODE

Choose und rule werden nicht mehr unterstützt, weil der Regelbasierte Optimizer desupported wurde. Inoffiziell können diese Optionen allerdings noch verwendet werden.

- STATISTICS_LEVEL

Es gibt 3 neue Statistik Level. Basic, typical und all. Der Stand der aktuellen Statistiken steht in der View V\$STATISTICS_LEVEL. Weitere Informationen hierzu sind im Kapitel Wartungsarbeiten zu finden

Basic: Es werden keine Statistiken erzeugt

Typical: Generiert Statistiken für den Shared Pool, SQL Execution Memory (PGA) und Recovery Zeit Abschätzung. Der Parameter DB_CACHE_ADVICE wird damit ungültig in 9.2.

All: Wie Typical + Row Source Execution Statistiken + Timed OS Statistiken

- PGA_AGGREGATE_TARGET

Pga_aggregate_target muß gesetzt sein. Wenn in ORACLE 10g PGA_AGGREGATE_TARGET nicht gesetzt wird, werden 20% der SGA für die PGA reserviert

- SHARED_POOL_SIZE

Bisher wurde von Oracle automatisch ein Overhead auf den vom Benutzer angegebenen Parameterwert hinzuaddiert. Ab Version 10g muss dieser vom Administrator mit eingerechnet werden.

- SGA_TARGET

SGA_TARGET wird im Zusammenhang mit Automatic Memory - Management verwendet.

Gesamtgröße, die die SGA benutzen darf. Bei diesem Parameter werden dann automatisch die folgenden Parameter gesetzt:

- db_cache_size
- shared_pool_size
- large_pool_size
- java_pool_size
- streams_pool_size

Der Selbsttuning Mechanismus ist nicht verfügbar für die folgenden Parameter:

- log_buffer
- db_keep_cache_size
- db_recycle_cache_size

- RECYCLEBIN

Defaultwert: ON

ON: Tabellen werden nach dem Löschen in den Recyclebin gelegt

OFF: Tabellen werden beim DROP sofort gelöscht.

Hinweis: Das Erzeugen von Statistiken mit `Analyze Schema` kann bei einer Oracle 10g R1 zu einem Fehler "ORA-38301: DDL/DML kann nicht über Objekten in Recycle-Bin ausgeführt werden" führen. Spätestens dann sollten die Schema Statistiken mit `DBMS_STATS.GATHER_SCHEMA_STATS` erzeugt werden.

Hinweis: Folgende Parameter sind Empfehlungen. Einige der Parameter sind Rechner (Ressourcen) abhängig. Zum Beispiel die des physikalischen Hauptspeichers. Daher können die hier aufgeführten Werte nur als Richtwerte dienen. Ähnlich dem Buffer Cache, ist der gemeinsame Pool erst durch genaue Messungen auf Ihrer Umgebung optimierbar.

Hinweis: Der Parameter ‚session_cached_cursors‘ ist ebenfalls erst durch genauere Messungen optimierbar. Die Idee der Session Cached Cursor ist es, Cursor für Befehle, die immer wiederkehrend aus einer Anwendung geöffnet werden, im Cache des einzelnen Prozesses zu halten. Der Parameter legt dabei die Anzahl der Cursor fest, die auf Prozessebene zur Verfügung stehen.

Hinweis: Automatic Memory Management (AMM) ist ein neues Feature in Oracle 10g. In Oracle 9i war es erstmals möglich, SGA-Parameter dynamisch, d.h. ohne Neustart der Datenbank, zu ändern. Die Änderungen mussten allerdings durch den DBA erfolgen. In ORACLE 10g überwacht die Datenbank automatisch die wechselnden Anforderungen an die einzelnen Bereiche der SGA und weist Hauptspeicher basierend auf den aktuellen Anforderungen zu. Bei ‚Automatic Shared Memory Management‘ handelt es sich um ein und dasselbe Feature.

Wenn Sie AMM verwenden, müssen Sie nur drei Initialisierungsparameter spezifizieren:

- SGA_TARGET ist ein neuer Parameter in ORACLE 10g. Er reflektiert die Gesamtgröße der SGA für AMM.
- SGA_MAX_SIZE bildet die statische Obergrenze der SGA
- PGA_AGGREGATE_TARGET definiert den reservierten Hauptspeicher für Sortierungen und Hash Joins. PGA_AGGREGATE_TARGET soll die Diskussionen um das Setzen der Parameter SORT_AREA_SIZE, HASH_AREA_SIZE usw. beenden. Man gibt dem System eine Obergrenze für alle PGA's aller Sitzungen vor; die Platzverteilung findet dann Lastabhängig statt.

Normalerweise besitzen SGA_TARGET und SGA_MAX_SIZE dieselbe Größe. Wenn Sie den Parameter SGA_MAX_SIZE höher setzen, reservieren Sie einen Puffer für Arbeitsspitzen der Datenbank. Nach dem Start der Instanz kann SGA_TARGET dynamisch maximal bis zur Größe von SGA_MAX_SIZE erhöht werden. Dagegen kann SGA_MAX_SIZE nicht ohne Neustart der Datenbank verändert werden.

Hinweis: Folgende Einstellungen werden für den Betrieb der ORACLE Instanz in ORACLE 10g R2 empfohlen. Ein Teil dieser Parameter wird obsolet, wenn AMM verwendet wird und wird daher nicht mehr aufgeführt (*db_cache_size*, *shared_pool_size*, *java_pool_size*). Für den Enterprise Manager alleine werden 80MB der *shared pool size* benötigt.

sga_max_size = Dieser Wert muß möglicherweise vergrößert werden, wenn die Werte für die *db_cache_size*, *shared_pool_size*, *log_buffer* usw. nicht mehr gesetzt werden können, da ein unzureichender Hauptspeicherspeicher reserviert wurde.

#Cache and I/O

db_block_size = 8192

*# db_block_size * db_file_multiblock_read_count* = 64kb

db_file_multiblock_read_count = 8

#Pools

java_pool_size = 0 # (min. 24MB, wenn JVM genutzt wird)

#Optimizer

optimizer_mode = FIRST_ROWS

#Processes and Sessions

processes = 400

#Cursors

open_cursors = 300

session_cached_cursors = 50

SMALL

#Pools

log_buffer = 2097152 # 2MB main memory

Sort, Hash Joins, Bitmap Indexes

pga_aggregate_target = 104857600 # 100MB main memory

LARGE**#Pools**

log_buffer = 3145728 # 3MB main memory

Sort, Hash Joins, Bitmap Indexes

pga_aggregate_target = 157286400 # 150MB main memory

System Managed Undo and Rollback Segments

undo_management = AUTO

undo_retention = 14400

#Cursors

cursor_sharing = SIMILAR

Miscellaneous

compatible = 10.2.0.1.0

Indizes

Bei entsprechender Konfiguration, Skripten oder Benutzung des Systems(z. B. Suchen nach einem bestimmten Attribut im DELMIA Process Engineer Finder), könnten die von DELMIA ausgelieferten Indizes nicht ausreichend sein, um einen linearen Zusammenhang von Datenvolumen und Laufzeit zu verhindern.

In diesem Fall müssen Sie eigene Indizes definieren. Diese Indizes werden beim nächsten Upgrade auf ein neues Release gelöscht. Halten Sie sich dazu bitte unbedingt an folgende Anleitung:

1. PPR Server anhalten

2. Index setzen

- ➊ Für einen Index auf der ORACLE Tabelle XDOX und der Spalte M_Y wählen Sie eine Nummer N aus, und vergeben Sie einen neuen Namen für Ihren Index. Der Index darf nicht mit dem Prefix „IDX“ beginnen.
- ➋ Führen Sie dieses Statement als Datenbankbenutzer e5_database aus.
- ➌ Möglicherweise müssen Sie den Namen des Tablespaces für die Indizes anpassen.

```
DROP INDEX CI_<N>;
CREATE INDEX CI_<N> ON XDOX ( M_Y ASC ) TABLESPACE ' ERGO_IDX' ;
```

3. Ermitteln Sie die Klassennummer:

```
SELECT CLASS_ID FROM POET_CLASSES WHERE TABLE_NAME = ' XDOX' ;
```

4. Bei folgendem Statement werden Sie nach der Klassennummer gefragt.

```
DELETE FROM POET_INDEXES WHERE SQL_INDEX_NAME = ' CI_<N>' ;
INSERT INTO POET_INDEXES(CLASS_ID, POET_INDEX_NAME,
SQL_INDEX_NAME, TABLE_NAME)
VALUES(&CLASS_ID, ' CI_<N>', ' CI_<N>', ' XDOX' );
COMMIT WORK;
```

5. Beispiel für Tabelle XDOERGOCOMPPRODUCTDEFAULTC242V0 und Spalte M_STRATTRIBUTE_40178V0:

```
CREATE INDEX CI_1 ON XDOERGOCOMPPRODUCTDEFAULTC242V0
(M_STRATTRIBUTE_40178V0 ASC ) TABLESPACE ' ERGO_IDX' ;
INSERT INTO POET_INDEXES(CLASS_ID, POET_INDEX_NAME,
SQL_INDEX_NAME, TABLE_NAME) VALUES(242, ' CI_1', ' CI_1',
' XDOERGOCOMPPRODUCTDEFAULTC242V0' );
COMMIT WORK;
```

Hinweis: Das Löschen von Indices durch den Upgrade Prozess kann man verhindern, indem man sich an eine bestimmte Namenskonvention hält. Indices mit dem Prefix '**CI_**' werden durch den Upgrade nicht gelöscht. Allerdings muss Kundenseitig sichergestellt werden, dass der Index weiterhin den gewünschten Effekt hat. Es besteht die Möglichkeit, dass ein Index nicht mehr notwendig ist oder unter Umständen sogar einen negativen Effekt hat.

Export und Import

Die Daten der Datenbank lassen sich mit ORACLE exportieren und importieren.

- ⇒ Ein Export ist lediglich eine Ergänzung zur physischen Sicherung. Dennoch unterstützt ein Export und Import der Daten die Migration zwischen den Versionen ORACLE 9i und ORACLE 10g.

Exportieren von Daten aus der Datenbank

- ➊ Starten Sie das Kommandofenster auf Ihrem Betriebssystem:
 - Start->Run->'cmd'
- ➋ Für die Datenbankbenutzer e5_database setzen Sie den Zeichensatz:
 - SET NLS_LANG=AMERICAN_AMERICA.UTF8
- ➌ Führen Sie den Export für e5_database für die Datenbankbenutzer durch, und geben Sie den Ort (im Beispiel c:\temp) für den Dump und für die Log-Datei an:
 - exp e5_database/databaseora file=c:\temp\e5_database.dmp
log=c:\temp\exp_e5_database.log
- ➍ Führen Sie danach den Export durch, und legen Sie Speicherplatz für das Dumpfile und das Logfile an:

Hinweis: Die Parameter ‚file‘ und ‚log‘ dienen zur Angabe des Dateinamens der zu erzeugenden Datei und zur Angabe einer Protokoll-Datei in die ein Protokoll des Exports abgelegt wird.

Hinweis: Für die Versionen vor Release 15 muss zusätzlich noch eine Sicherung des Users e5_base durch das folgende Kommando durchgeführt werden: exp e5_base/baseora file=c:\temp\e5_base.dmp log=c:\temp\exp_e5_base.log

Importieren von Daten in die Datenbank

Vor dem Importieren der Daten in die Datenbank, muss die Datenbank mit der entsprechenden physischen und logischen Struktur (Datenbanksegmente, Datenbankdateien) angelegt worden sein. Dies ist in der Regel die Aufgabe eines Datenbankadministrators.

Die von uns mitgelieferte **ORACLE 10g Setup DVD** erlaubt allerdings die Erzeugung einer leeren Datenbank. Vor dem Importieren der Daten in die Datenbank, müssen außerdem die passenden Benutzer angelegt werden.

Ist die Datenbank mit der von uns mitgelieferten ORACLE 10g Setup CD erstellt, dann wird jedem erzeugten User automatisch der *Tablespace temp* als Default Tablespace zugewiesen. Daher muss dem *CREATE USER statement* in diesem Fall kein Default *Tablespace* mehr zugewiesen werden.

Falls Sie Ihre Datenbank auf eine andere Art erzeugen, ist es unbedingt notwendig den temporären Daten einen *Tablespace* zuzuweisen. Seit ORACLE 10g gibt es die *Default Permanent Tablespace*. Diese gilt für die gesamte Datenbank und wird verwendet, wenn einem Benutzer keine Default Tablespace zugewiesen wurde.

Nach dem Anlegen der User, ist es notwendig, den Inhalt der Umgebungsvariable `NLS_LANG` korrekt zu setzen.

Mit der V5 R14 Release wurden Datentypen vom Typ *long raw* in den Datentyp *blob* *konvertiert*. Dadurch werden Erweiterungen der Import Skripte notwendig. Vor dem Import sind daher die Blob Tabellen zu erzeugen. Nähere Informationen sind dem Kapitel **Blob Tabellen erzeugen**, zu entnehmen.

Der eigentliche Import wird dann mit dem `imp`-Kommando durchgeführt:

- `SET NLS_LANG=AMERICAN_AMERICA.UTF8`
- `imp system/syselcaro fromuser=e5_database touser =e5_database
file=e5_database.dmp log=import_e5_database.log`

Import der Process Engineer Datenbank

Benutzerschema

Für den PPR Hub wird ein Datenbankbenutzer benötigt. Die Namen und die Passwörter können vom DB-Administrator frei gewählt werden. Die Standardnamen und -passwörter von DELMIA sind:

- e5_database/databaseora

neue Benutzer in einer Datenbank anlegen zu können, muss eine Anmeldung über einen User erfolgen, der über die erforderlichen Rechte verfügt:

- Benutzername: sys
- Passwort: syselcaro oder change_on_install

Hinweis: Falls keines der beiden Passwörter angenommen wird und die ORACLE - Installation nicht neu erstellt wurde, fordern Sie vom Datenbankadministrator die Passwörter für die **User SYS** und **SYSTEM** an, oder lassen Sie die nächsten Schritte vom Administrator ausführen.

Die folgenden Kommandos erzeugen den e5_database-User, der die Daten der Konfiguration von DPE enthält:

Zuerst werden die alten Rollen und User gelöscht. Ab Release R15 können die Kommandos ignoriert werden, die sich auf den User e5_base beziehen.

Rollen:

```
drop role e5_base_user;
drop role e5_database_user;
```

User:

```
drop user e5_base cascade;
drop user e5_database cascade;
```

```
CREATE USER e5_database IDENTIFIED BY databaseora
DEFAULT TABLESPACE ERGO_USR TEMPORARY TABLESPACE TEMP;
```

```
SET NLS_LANG=AMERICAN_AMERICA.UTF8
```

Berechtigungen

Die Datenbankbenutzer benötigen folgende Berechtigungen in der Datenbank:

- grant alter session
- grant create cluster
- grant create database link
- grant create procedure
- grant create sequence
- grant create session
- grant create synonym
- grant create table
- grant create trigger
- grant create view

Zusätzlich müssen zwei Select-Rechte für den neu erzeugten User mittels der Kommandos vergeben werden. Diese beiden Berechtigungen können nur vom Systemadministrator vergeben werden:

- grant select on v_\$process
- grant select on v_\$session

Für den CREATE USER- Befehl müssen Sie das CREATE USER- Privileg besitzen, das Sie mit der DBA-Rolle automatisch haben. Das CREATE USER-Privileg ist sehr mächtig. Achten Sie darauf, dass möglichst wenige Benutzer dieses Privileg besitzen.

Privilegien können einem Benutzer direkt oder über Rollen zugewiesen werden. Rollen können Privilegien und andere Rollen enthalten.

Alternativ können Sie die Verwaltung von Benutzern und Rollen auch über den Enterprise Manager vornehmen.

Hinweis: Um Probleme im Zusammenhang mit Stored Procedures und Grants zu vermeiden, sollte *execute immediate* in Stored Procedures vermieden werden.

Tablespaces für Benutzer

Das Anlegen der Benutzer umfasst auch die Definition der Tablespaces.

Die Datenbankbenutzer sollten *den StandardTablespace* ERGO_USR und den temporären Tablespaces TEMP (abhängig von den im Kapitel „Tablespace“ vergebenen Namen) benutzen.

Weiterhin sollten folgende **Quotas** (Kontingente) gesetzt werden. Ein neu erstellter Benutzer hat von Haus aus keine Speicherplatzanteile an den Tablespaces. Diese Anteile werden Quotas genannt. (beispielhaft für den Benutzer „e5_database“):

```
alter user e5_database quota unlimited on ERGO_INDX;  
alter user e5_database quota unlimited on ERGO_USR;  
alter user e5_database quota 0K on system;
```

Zu beachten ist hierbei, dass die Tablespaces *ergo_indx* und *ergo_usr* existieren müssen. Falls die ORACLE-Instanz, die die DPE-User aufnehmen soll, nicht mittels eines von DELMIA zur Verfügung gestellten Setups erstellt wurde, müssen evtl. Anpassungen bei der Benennung vorgenommen werden.

Benutzer benötigen keine Quota für Temporary Tablespaces und Undo Tablespaces. Diese Segmente können ohne zusätzliche Privilegien erstellt werden und werden von ORACLE verwaltet.

Rollen anlegen

Die Rolle CONNECT enthält das CREATE SESSION –Privileg. Nachdem der Benutzer die Rolle CONNECT erhalten hat, kann er sich anmelden. Jetzt müssen noch Rollen für die verschiedenen User erzeugt werden. Diese geschieht durch das Ausführen des SQL-Kommandos:

```
connect e5_database/databaseora  
  
CREATE ROLE E5_DATABASE_USER NOT IDENTIFIED
```

Nach dem Anlegen der User, ist es notwendig, den Inhalt der Umgebungsvariable NLS_LANG korrekt zu setzen.

Blob Tabellen erzeugen

Mit der V5 R14 Release wurden Datentypen vom Typ *long raw* in den Datentyp *blob* *konvertiert*. Dadurch werden Erweiterungen der Import Skripte notwendig.

Folgende Blob- Tabellen müssen vor dem Import für den User „e5_database“ erzeugt werden:

```
"XBLOBVALUEC497V11", "XDOSCRIPTC339V0", „XDOPRINTFORMC474V8" ,  
"XDOSCRIPTVARIABLEC473V8", "XDOVBAPROJECTC466V8", "XMURULE"
```

Die DDL- Kommandos für den Benutzer “e5_database“ ist dem Anhang zu entnehmen.

Sind die Tabellen angelegt, kann auf DOS-Ebene der Import der Datenbanken durchgeführt werden. Zu beachten ist hierbei, dass der Parameter *ignore* auf *yes* gesetzt sein muss (*ignore=y*).

Data-Pump

Data-Pump ist ein neues Werkzeug für das Exportieren und Importieren von Daten. Es ist hervorragend für logische Datensicherungen und den Transport von Tabellen oder ganzen Schemas zwischen Datenbanken geeignet. Die Steuerungsprogramme heißen expdp und impdp. Aus Kompatibilitätsgründen werden weiterhin das alte Export (exp) und Import (imp) Tool ausgeliefert.

Vorteile:

- Serverbasiertes Werkzeug zum schnellen Laden und Entladen von Daten und Metadaten.
- Aufrufbar über PL/SQL
- Webbasiert (EM DB-Control, EM Grid-Control)

Der Data-Pump Mechanismus basiert auf einem speziellen Job Konzept, welches nur direkt auf dem ORACLE Server lauffähig ist. Dadurch kann der Datentransfer nur noch direkt auf dem ORACLE Server durchgeführt werden. Bei der Verwendung von Data-Pump können verschiedene Jobs im Hintergrund abgesetzt werden. Diese können bei einem Abbruch auch wieder fortgesetzt werden.

Aufgrund der Aufteilung des Exports und Imports in verschiedene Prozesse besteht außerdem die Möglichkeit der Parallelisierung.

Eine graphische Oberfläche zur Vereinfachung der Handhabung des Data-Pump Mechanismus stellt ORACLE webbasiert über den Enterprise Manager zur Verfügung.

Hinweise zur Bedienung

Export

Der Enterprise Manager bietet die Möglichkeit die Datenbank über den Menüpunkt *Wartung* zu exportieren. Wählen Sie hierzu den Data-Pump-Export über den Link *IN DATEIEN EXPORTIEREN*.

Über die Option *Schema* auf der nächsten Seite können eine oder mehrere Schemas und die darin enthaltenen Objekte exportiert werden. Auf diese Weise besteht dann die Möglichkeit den E5 Datenbankbenutzer auszuwählen, der exportiert werden soll.

Weitere Details und das Konfigurieren der Parameter für das Verzeichnisobjekt oder die Log-Datei entnehmen Sie bitte der einschlägigen ORACLE Literatur. Über den Link *ERWEITERTE OPTIONEN ANZEIGEN* können Sie weitere Features der ORACLE Data-Pump nutzen.

Import

Ebenso wie für den Export besteht über den ORACLE Enterprise Manager die Möglichkeit Datenbank Schemas zu importieren. Auch beim Data-Pump-Import startet man über den Menüpunkt *Wartung*. Aus den verfügbaren Dienstprogrammen muss jetzt *AUS DATEIEN IMPORTIEREN* gewählt werden. Falls man in der Auswahlliste *DATENBANKVERSION VOR 10g* angibt, führt der ORACLE Enterprise Manager einen Import mit einer Exportdatei durch, die durch das bisherige Export Steuerungsprogramm exp erstellt wurde.

Als *Importtyp* muss jetzt die Option *Ganze Dateien* gewählt werden. Es besteht auf weiteren Seiten z.B. Über *Erneute Zuordnung*, die Möglichkeit Daten von einem in einen anderen Tabellenbereich zu importieren. Damit wird das explizite Erzeugen von BLOB- Tabellen, wie es in früheren Importskripten oder über den dbAssistant notwendig war, hinfällig.

Existiert ein E5 Datenbankbenutzer bereits, dann wird in den bestehenden importiert, ansonsten wird er automatisch erzeugt.

Beachten Sie bitte bei der Verwendung der erweiterten Optionen die Hinweise in der originalen ORACLE Dokumentation.

Hinweis: Auf die oben aufgeführten Möglichkeiten näher einzugehen würde an dieser Stelle den Rahmen dieser Dokumentation sprengen. Jedoch haben hier die allgemein von ORACLE empfohlenen Vorschläge Gültigkeit, die in der einschlägigen Literatur unter der Rubrik ‚ORACLE Datenbank-Administration‘ zur Verfügung stehen. .

Process Engineer Datenbank

Wenn Sie noch nicht mit dem *ENOVIA Manufacturing Hub* gearbeitet haben, können Sie zuerst einmal mit der Process Engineer Datenbank beginnen.

In dieser Datenbank existieren einfache Beispielprojekte, die Sie sofort anwenden können. Die Process Engineer Datenbank importieren Sie in neun einfachen Schritten:

Import einer ORACLE (Demonstrations) Datenbank:

- Starten Sie das Tool DBAssistant.exe im Verzeichnis DELMIA\PPRServer\program\bin.
- Wählen Sie den Punkt ‚Exchange – Import‘ und drücken Sie ‚OK‘.

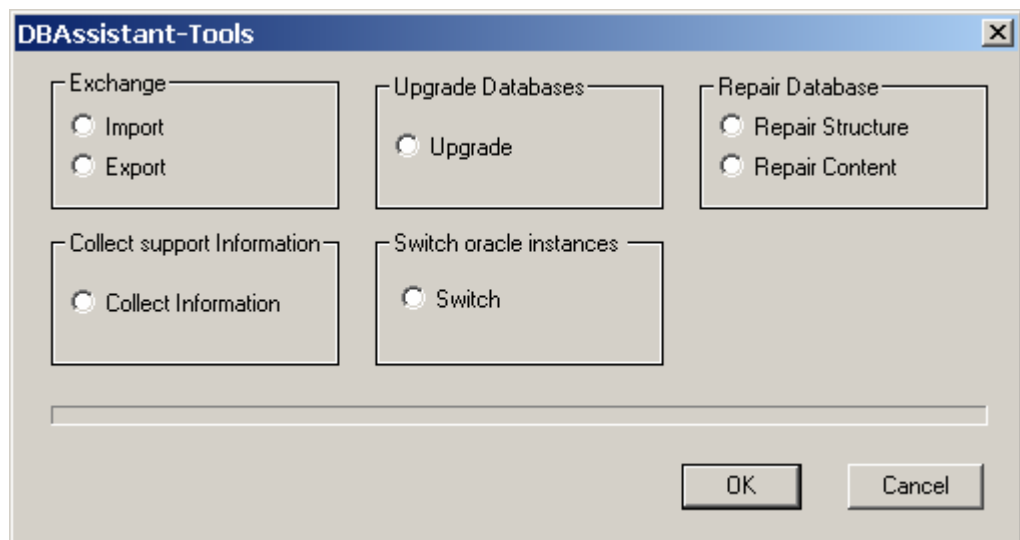
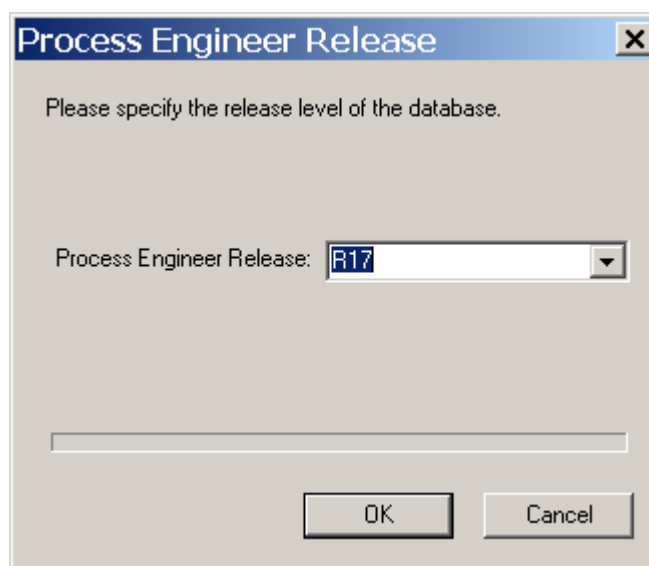


Abbildung 1: Import der Process Engineer Datenbank



- ➊ Geben Sie die Verbindungsparameter für die ORACLE Datenbank ein.
- ➋ Geben Sie dabei auch die Benutzer an, die erzeugt werden sollen.
- ➌ Wählen Sie als zu importierende Datenbank das Verzeichnis „Oracle-Dumps“ auf der CD aus. Falls Sie die Process Engineer Datenbank importieren möchten, das Unterverzeichnis „...\\Oracle-Dumps\\pe517“.

The screenshot shows the 'ImportAssistant - Tool' window. It contains several input sections:

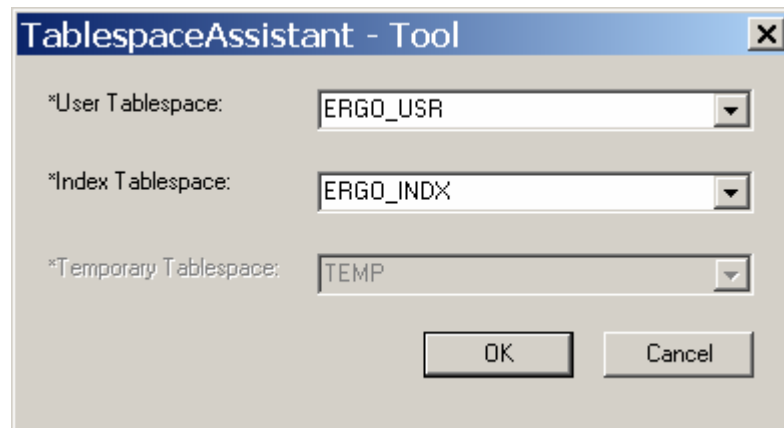
- DB-Database:** *OriginUsername: e5_database, *DestinationUsername: e5_database, *DestinationPassword: databaseora, *Filename: C:\\program files\\R17\\ (with a Browse button).
- DB-Base:** *OriginUsername, *DestinationUsername, *DestinationPassword, *Filename (all empty, with a Browse button).
- DB-Ergotime:** OriginUsername, DestinationUsername, DestinationPassword, Filename (all empty, with a Browse button).
- DB-Admin-Accounts:** *SysUsername: sys, *SysPassword: syselcaro, *SystemUsername: system, *SystemPassword: syselcaro.
- DB-Parameter:** Two checked checkboxes: 'Use standard Tablespacenames' and 'Use Sysdba'.
- DB-Servicename:** DBServicename (empty).

At the bottom right are 'OK' and 'Cancel' buttons.

Abbildung 2: Beispielhafte Importparameter für die Process Engineer Datenbank

- ➍ Wurde beim Export ein Name für einen Blob Dump vergeben und wird beim Import nicht der Standard- Tablespacenames verwendet, dann ist die Checkbox für die Verwendung von Standard Tablespacenames zu deaktivieren.
- ➎ Drücken Sie ‚Ok‘.

- ➊ Es erscheint ein weiterer Dialog in dessen Eingabefelder müssen die entsprechenden Tablespaces mit angegeben werden.



- ➋ Drücken Sie erneut ,Ok'.

Upgrade einer vorhanden Datenbank

Wenn Sie bisher ORACLE 9i eingesetzt haben, lesen Sie bitte die beiden Kapitel ‚Vorraussetzungen für die Installation‘ und ‚Migrieren einer ORACLE 9i Datenbank‘.

Für den Upgrade einer vorhandenen Datenbank ist auf Datenbankseite nichts zu beachten, der Setup des PPR Servers wird selbstständig alle notwendigen Änderungen am Schema vornehmen.

Verwendung des Datenbankassistenten von DPE

Neben der Möglichkeit die Benutzer manuell anzulegen und die Datenbankimporte Kommandozeilengestützt durchzuführen kann, wie auch am Beispiel des Imports der Process Engineer Datenbank zu sehen ist, auch ein GUI-basiertes Tool eingesetzt werden, um dieses Aufgaben durchzuführen.

Hierzu kann im Start-Dialog der DPE-Installations-CD der Menüpunkt Datenbankassistent aufgerufen werden.

In einem Dialog kann dann ausgewählt werden, ob ein Im- oder Export, ein Datenbankupgrade, ein Sammeln support-relevanter Informationen, oder beispielsweise das Umschalten des Datenbank-Servers ausgeführt werden soll.

Im Kapitel **Process Engineer Datenbank** ist ein Beispiel für den Import von Dumps zu finden.

Hinweise zur Wartung der Datenbank

Performance-Tuning

Optimierung der Datenbank-Instanz

Einige der Parameter aus dem Kapitel Instanzeinstellungen sind Rechner (Ressourcen) abhängig, zum Beispiel die des physikalischen Hauptspeichers. Daher können die hier aufgeführten Werte nur als Richtwerte dienen.

Ähnlich dem Buffer Cache (`db_cache_size`), ist der gemeinsame Pool (`shared_pool_size`) erst durch genaue Messungen auf Ihrer Umgebung optimierbar. Beide Größen können ab der ORACLE 9i bei laufendem Betrieb angepasst werden. Weitere relevante Parameter für die Auslastung des physikalischen Hauptspeichers sind die Sort Area (`sort_area_size` – ORACLE 8i) der Log Buffer (`log_buffer`). Wird mit Automatic Memory Management (AMM) gearbeitet gilt folgendes:

Hinweis: Automatic Memory Management (AMM) ist ein neues Feature in Oracle 10g. In Oracle 9i war es erstmals möglich, SGA-Parameter dynamisch, d.h. ohne Neustart der Datenbank, zu ändern. Die Änderungen mussten allerdings durch den DBA erfolgen. In ORACLE 10g überwacht die Datenbank automatisch die wechselnden Anforderungen an die einzelnen Bereiche der SGA und weist Hauptspeicher basierend auf den aktuellen Anforderungen zu. Bei ‚Automatic Shared Memory Management‘ handelt es sich um ein und dasselbe Feature.

Dynamisch konfigurierbare Parameter lassen sich mittels ALTER SYSTEM SET <name>=<new value> ändern.

Es wird immer das ganze Vielfache eines so genannten **Granules** allokiert. Der neue Parameter SGA_MAXSIZE legt die maximale Gesamtgröße der SGA fest. Diese darf durch die Summe der Buffer sowie aller weiteren allokierten Bereiche in der SGA nicht überschritten werden.

Durch Setzen des Parameters `optimizer_mode` auf den Wert `first_rows` wird eine Mischung aus kostenbasierten und heuristischen Analyseverfahren eingesetzt, um die in der Datenbank ablaufenden SQL-Kommandos zu optimieren.

Das Setzen dieses Parameters kann entweder durch einen Eintrag in der initial ausgeführten Steuerungstabelle der Datenbank (`init/INSTANZNAME.ora`) oder durch das SQL-Statement

```
ALTER SESSION SET OPTIMIZER_MODE=first_rows
```

 erfolgen.

Genauere Informationen über die Funktionsweise des Optimierers und den Auswirkungen dieses Parameters können dem ORACLE-Handbuch

[‚ORACLE 10g Database Performance Guide and Reference‘](#)

entnommen werden.

Nach dem Setzen dieses Parameters ist eine regelmäßige Kontrolle der Schemas des Datenbankusers (`e5_database`) notwendig.

Wartungsarbeiten

Einfache Überwachungstätigkeiten

Als Minimum empfiehlt es sich mindestens den Hauptspeicherverbrauch und den freien Plattenplatz auf dem Datenbankserver in regelmäßigen Abständen zu kontrollieren. Treten hier Engpässe auf, können als Folge massive Performanceeinbußen auftreten, die auch zu Programmabstürzen führen können.

- Die Datenbank ist so konfiguriert, dass sie, falls ein höherer Platzbedarf erforderlich ist, automatisch die physikalischen Datenbank-Dateien vergrößert.
- Während der physikalisch benötigte Speicherplatz auf dem Dateisystem hauptsächlich von der verarbeiteten Datenmenge abhängt, wird die Menge an verwendetem Hauptspeicher größtenteils durch die gleichzeitig angemeldete Anzahl von Benutzern und die Art der Applikation bestimmt, die auf Datenbank zugreift. Ergeben sich hier im laufenden Betrieb größere Änderungen, so ist eine Neukonfiguration der Datenbankinstanz erforderlich.
- Für eine optimale Ausführung der generierten SQL Statements ist es wichtig, in regelmäßigen Abständen Statistiken zu erstellen. Diese können für Tabellen, Indexes und Felder erstellt werden. An dieser Stelle möchten wir auf den 'Performance Tuning Guide' verweisen.
- Nichtsdestotrotz soll an dieser Stelle auf ein paar elementare Dinge hingewiesen werden, die sich bei der ORACLE 10g R2 Release, gegenüber den Vorgängerversionen unterscheiden. In ORACLE 10g existieren eine Menge zusätzliche Advisories für unterschiedliche Zwecke. Sie sammeln Informationen und analysieren Statistiken und unterbreiten auf dieser Basis Vorschläge für die Verbesserung von Performance oder die Beseitigung von Problemen. Die Advisories benutzen das Automatic Workload Repository und beziehen sich in der Regel auf den aktuellen Workload.

OPTIMIZER_MODE:

Choose und rule werden nicht mehr unterstützt, weil der Regelbasierte Optimizer nicht mehr supported wird. Inoffiziell können diese Optionen allerdings noch verwendet werden. Durch Verwendung der ORACLE 10g sind punktgenauere Analysen möglich. Statistiken werden automatisch gesammelt und

- Im AWR (Automatic Workload Repository) abgelegt
- Das ASH (Active Session History) gibt Informationen über Sessions wieder. Es werden historische Informationen zur Sitzungshistorie einer Instanz zur Verfügung gestellt.
- Der ADDM (Automatic Database Diagnostic Monitor) wertet all das aus und gibt Empfehlungen

Der Fokus liegt auf der gesamten Datenbank. Es werden jedoch keine speziellen Sessions oder Applikationen angepasst.

Voraussetzungen für das ‚Automatische SGA Tuning‘ sind folgende:

- Der STATISTICS_LEVEL muss auf TYPICAL oder ALL stehen

Damit ist das ‚Automatic SGA Tuning‘ möglich und Systemstatistiken werden alle 60 Minuten gesammelt

Wer diese Funktionalität gezielt abschalten will, stellt das Statistikintervall ab oder kann die ACTIVE SESSION HISTORY abschalten.

Wenn der STATISTICS_LEVEL auf Basic steht ist das

- Tabellen-Monitoring abgestellt
- Automatic-SGA-Management geht nicht
- SGA und PGA-Advisors funktionieren nicht

Hinweis: Auf die oben aufgeführten und weiteren Möglichkeiten näher einzugehen würde an dieser Stelle den Rahmen dieser Dokumentation sprengen. Jedoch haben hier die allgemein von ORACLE empfohlenen Vorschläge Gültigkeit, die in der einschlägigen Literatur unter der Rubrik ‚ORACLE Datenbank-administration‘ zur Verfügung stehen. .

Anhang

Ergänzend zum Kapitel ‚Blob Tabellen erzeugen‘

Die DDL- Kommandos zum erzeugen der Blob – Tabellen für den Benutzer „e5_database“:

```
CREATE TABLE "XBLOBVALUEC497V11"
("OID" NUMBER(38, 0) DEFAULT 0 NOT NULL ENABLE,
"CID" NUMBER(38, 0) DEFAULT 0 NOT NULL ENABLE,
"M_DATA0V13" BLOB DEFAULT EMPTY_BLOB(),
PRIMARY KEY ( OID ) )
/

CREATE TABLE "XDOSCRIPTC339V0" (
"OID" NUMBER(38, 0) DEFAULT 0 NOT NULL ENABLE,
"CID" NUMBER(38, 0) DEFAULT 0 NOT NULL ENABLE,
"M_PCHANGEPROTOCOLOID0V0" NUMBER(38, 0) DEFAULT 0,
"M_PCHANGEPROTOCOLCID1V0" NUMBER(38, 0) DEFAULT 0,
"M_NAME2V0" VARCHAR2(1000),
"M_NAMESHORT3V0" VARCHAR2(1000),
"M_NOTE4V0" VARCHAR2(4000),
"M_TYPENAME5V0" VARCHAR2(1000),
"M_TABLENAME6V0" VARCHAR2(1000),
"M_STREXTERNALID7V0" VARCHAR2(1000),
"M_CREATIONDATE8V0" DATE,
"M_MODIFICATIONDATE9V0" DATE,
"M_BWRITECHANGEPROTOCOL10V0" CHAR(1) DEFAULT chr(0),
"M_DBLSORTINDEX12V0" NUMBER DEFAULT 0,
"M_STRCREATOR13V0" VARCHAR2(1000),
"M_STRMODIFIER14V0" VARCHAR2(1000),
"M_PDODEFAULTTIMPLOID37V0" NUMBER(38, 0) DEFAULT 0,
"M_PDODEFAULTTIMPLCID38V0" NUMBER(38, 0) DEFAULT 0,
"M_TYPE39V0" VARCHAR2(1000),
"M_USEGRID41V0" CHAR(1) DEFAULT chr(0),
"M_STROBJECTUUID8V3" VARCHAR2(1000),
"M_RUNASOWNER53V4" CHAR(1) DEFAULT chr(0),
"M_STRORIGINALUUID9V8" VARCHAR2(1000),
"M_PUSEROWNEROID20V8" NUMBER(38, 0) DEFAULT 0,
"M_PUSEROWNERCID21V8" NUMBER(38, 0) DEFAULT 0,
"M_RUNOWNTRANSACTION74V8" CHAR(1) DEFAULT chr(0),
"M_STROEMOBJECTUUID10V13" VARCHAR2(1000),
```



```

"M_SCRIPT24V13" BLOB DEFAULT EMPTY_BLOB(),
"STRPLMID" VARCHAR2(1000),
"BLOBNOTEID" NUMBER(38, 0) DEFAULT 0,
"BLOBNOTEID" NUMBER(38, 0) DEFAULT 0,
"BSIMPLESETEXIST" CHAR(1) DEFAULT chr(0),
"PERGOITEMPARENTOID" NUMBER(38, 0) DEFAULT 0,
"PERGOITEMPARENTCID" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING1_STRATATTRIBUTE" VARCHAR2(4000),
"LARGESTRING1_BLOBATTRIBUTEID" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING1_BLOBATTRIBUTEID" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING2_STRATATTRIBUTE" VARCHAR2(4000),
"LARGESTRING2_BLOBATTRIBUTEID" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING2_BLOBATTRIBUTEID" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING3_STRATATTRIBUTE" VARCHAR2(4000),
"LARGESTRING3_BLOBATTRIBUTEID" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING3_BLOBATTRIBUTEID" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING4_STRATATTRIBUTE" VARCHAR2(4000),
"LARGESTRING4_BLOBATTRIBUTEID" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING4_BLOBATTRIBUTEID" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING5_STRATATTRIBUTE" VARCHAR2(4000),
"LARGESTRING5_BLOBATTRIBUTEID" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING5_BLOBATTRIBUTEID" NUMBER(38, 0) DEFAULT 0,
PRIMARY KEY ( OID ) )
/

```

```

CREATE TABLE "XDOPRINTFORMC474V8"
("OID" NUMBER(38, 0) DEFAULT 0 NOT NULL ENABLE,
"CID" NUMBER(38, 0) DEFAULT 0 NOT NULL ENABLE,
"M_PCHANGEPROTOCOLID0V0" NUMBER(38, 0) DEFAULT 0,
"M_PCHANGEPROTOCOLCID1V0" NUMBER(38, 0) DEFAULT 0,
"M_NAME2V0" VARCHAR2(1000),
"M_NAMESHORT3V0" VARCHAR2(1000),
"M_NOTE4V0" VARCHAR2(4000),
"M_TYPENAME5V0" VARCHAR2(1000),
"M_TABLENAME6V0" VARCHAR2(1000),
"M_STREXTERNALID7V0" VARCHAR2(1000),
"M_STROBJECTUUID8V3" VARCHAR2(1000),
"M_STRORIGINALUUID9V8" VARCHAR2(1000),
"M_CREATIONDATE8V0" DATE,
"M_MODIFICATIONDATE9V0" DATE,
"M_BWRITECHANGEPROTOCOL10V0" CHAR(1) DEFAULT chr(0),
"M_DBLSORTINDEX12V0" NUMBER DEFAULT 0,
"M_STRCREATOR13V0" VARCHAR2(1000),
"M_STRMODIFIER14V0" VARCHAR2(1000),

```

```

"M_PUSEROWNEROID20V8" NUMBER(38, 0) DEFAULT 0,
"M_PUSEROWNERCID21V8" NUMBER(38, 0) DEFAULT 0,
"M_PPLANTYPEOID22V8" NUMBER(38, 0) DEFAULT 0,
"M_PPLANTYPECID23V8" NUMBER(38, 0) DEFAULT 0,
"M_BDEFAULTFORM24V8" CHAR(1) DEFAULT chr(0),
"M_STROEMOBJECTUUID10V13" VARCHAR2(1000),
"M_FORMDATA23V13" BLOB DEFAULT EMPTY_BLOB(),
"STRPLMID" VARCHAR2(1000),
"BLOBNOTEOID" NUMBER(38, 0) DEFAULT 0,
"BLOBNOTECID" NUMBER(38, 0) DEFAULT 0,
"BSIMPLESETEXIST" CHAR(1) DEFAULT chr(0),
PRIMARY KEY ( OID ) )
/

```

```

CREATE TABLE "XDSCRIPTVARIABLEC473V8"
("OID" NUMBER(38, 0) DEFAULT 0 NOT NULL ENABLE,
"CID" NUMBER(38, 0) DEFAULT 0 NOT NULL ENABLE,
"M_PCHANGEPROTOCOLOID0V0" NUMBER(38, 0) DEFAULT 0,
"M_PCHANGEPROTOCOLCID1V0" NUMBER(38, 0) DEFAULT 0,
"M_NAME2V0" VARCHAR2(1000),
"M_NAMESHORT3V0" VARCHAR2(1000),
"M_NOTE4V0" VARCHAR2(4000),
"M_TYPENAME5V0" VARCHAR2(1000),
"M_TABLENAME6V0" VARCHAR2(1000),
"M_STREXTERNALID7V0" VARCHAR2(1000),
"M_STROBJECTUUID8V3" VARCHAR2(1000),
"M_STRORIGINALUUID9V8" VARCHAR2(1000),
"M_CREATIONDATE8V0" DATE,
"M_MODIFICATIONDATE9V0" DATE,
"M_BWRITECHANGEPROTOCOL10V0" CHAR(1) DEFAULT chr(0),
"M_DBLSORTINDEX12V0" NUMBER DEFAULT 0,
"M_STRCREATOR13V0" VARCHAR2(1000),
"M_STRMODIFIER14V0" VARCHAR2(1000),
"M_PUSEROWNEROID20V8" NUMBER(38, 0) DEFAULT 0,
"M_PUSEROWNERCID21V8" NUMBER(38, 0) DEFAULT 0,
"M_PPLANTYPEOID22V8" NUMBER(38, 0) DEFAULT 0,
"M_PPLANTYPECID23V8" NUMBER(38, 0) DEFAULT 0,
"M_LTYPE24V8" NUMBER(38, 0) DEFAULT 0,
"M_LDATATYPE26V8" NUMBER(38, 0) DEFAULT 0,
"M_STROEMOBJECTUUID10V13" VARCHAR2(1000),
"M_SCRIPT23V13" BLOB DEFAULT EMPTY_BLOB(),
"STRPLMID" VARCHAR2(1000),
"BLOBNOTEOID" NUMBER(38, 0) DEFAULT 0,
"BLOBNOTECID" NUMBER(38, 0) DEFAULT 0,

```

```

"BSIMPLESETEXIST" CHAR(1) DEFAULT chr(0),
"STRDSN" VARCHAR2 (1000),
"STRUID" VARCHAR2 (1000),
"BNOTVISIBLE" CHAR (1) DEFAULT chr(0),
"PScriptVARIABLEOID" NUMBER (38) DEFAULT 0,
"PScriptVARIABLECID" NUMBER (38) DEFAULT 0,
"PCONDITIONS" BLOB DEFAULT EMPTY_BLOB(),
PRIMARY KEY ( OID ) )
/

CREATE TABLE "XDOVBAPROJECTC466V8"
("OID" NUMBER(38, 0) DEFAULT 0 NOT NULL ENABLE,
"CID" NUMBER(38, 0) DEFAULT 0 NOT NULL ENABLE,
"M_PCHANGEPROTOCOLOID0V0" NUMBER(38, 0) DEFAULT 0,
"M_PCHANGEPROTOCOLCID1V0" NUMBER(38, 0) DEFAULT 0,
"M_NAME2V0" VARCHAR2(1000),
"M_NAMESHORT3V0" VARCHAR2(1000),
"M_NOTE4V0" VARCHAR2(4000),
"M_TYPENAME5V0" VARCHAR2(1000),
"M_TABLENAME6V0" VARCHAR2(1000),
"M_STREXTERNALID7V0" VARCHAR2(1000),
"M_STROBJECTUUID8V3" VARCHAR2(1000),
"M_STRORIGINALUUID9V8" VARCHAR2(1000),
"M_CREATIONDATE8V0" DATE,
"M_MODIFICATIONDATE9V0" DATE,
"M_BWRITECHANGEPROTOCOL10V0" CHAR(1) DEFAULT chr(0),
"M_DBLSORTINDEX12V0" NUMBER DEFAULT 0,
"M_STRCREATOR13V0" VARCHAR2(1000),
"M_STRMODIFIER14V0" VARCHAR2(1000),
"M_PUSEROWNEROID20V8" NUMBER(38, 0) DEFAULT 0,
"M_PUSEROWNERCID21V8" NUMBER(38, 0) DEFAULT 0,
"M_PDODEFAULTTIMPLOID37V0" NUMBER(38, 0) DEFAULT 0,
"M_PDODEFAULTTIMPLCID38V0" NUMBER(38, 0) DEFAULT 0,
"M_STROEMOBJECTUUID10V13" VARCHAR2(1000),
"M_DATA23V13" BLOB DEFAULT EMPTY_BLOB(),
"STRPLMID" VARCHAR2(1000),
"BLOBNOTEOID" NUMBER(38, 0) DEFAULT 0,
"BLOBNOTECID" NUMBER(38, 0) DEFAULT 0,
"BSIMPLESETEXIST" CHAR(1) DEFAULT chr(0),
"PERGOITEMPARENTOID" NUMBER(38, 0) DEFAULT 0,
"PERGOITEMPARENTCID" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING1_STRATTRIBUTE" VARCHAR2(4000),
"LARGESTRING1_BLOBATTRIBUTE" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING1_BLOBATTRIBUTE" NUMBER(38, 0) DEFAULT 0,

```

```

"LARGESTRING2_STRATTRIBUTE" VARCHAR2(4000),
"LARGESTRING2_BLOBATTRIBUTEID" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING2_BLOBATTRIBUTEID" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING3_STRATTRIBUTE" VARCHAR2(4000),
"LARGESTRING3_BLOBATTRIBUTEID" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING3_BLOBATTRIBUTEID" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING4_STRATTRIBUTE" VARCHAR2(4000),
"LARGESTRING4_BLOBATTRIBUTEID" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING4_BLOBATTRIBUTEID" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING5_STRATTRIBUTE" VARCHAR2(4000),
"LARGESTRING5_BLOBATTRIBUTEID" NUMBER(38, 0) DEFAULT 0,
"LARGESTRING5_BLOBATTRIBUTEID" NUMBER(38, 0) DEFAULT 0,
PRIMARY KEY ( OID ) )
/

```

```

CREATE TABLE "XMURULE"
("OID" NUMBER(38, 0) DEFAULT 0 NOT NULL ENABLE,
"CID" NUMBER(38, 0) DEFAULT 0 NOT NULL ENABLE,
"BLOBFIRSTCONDITION" BLOB DEFAULT EMPTY_BLOB(),
"IFIRSTRESULT" NUMBER(38, 0) DEFAULT 0,
"BLOBSECONDCONDITION" BLOB DEFAULT EMPTY_BLOB(),
"ISECONDRESULT" NUMBER(38, 0) DEFAULT 0,
"ITHIRDRESULT" NUMBER(38, 0) DEFAULT 0,
"M_PCHANGEPROTOCOLOID0V0" NUMBER(38, 0) DEFAULT 0,
"M_PCHANGEPROTOCOLCID1V0" NUMBER(38, 0) DEFAULT 0,
"M_NAME2V0" VARCHAR2(1000),
"M_NAMESHORT3V0" VARCHAR2(1000),
"M_NOTE4V0" VARCHAR2(4000),
"BLOBNOTEID" NUMBER(38, 0) DEFAULT 0,
"BLOBNOTEID" NUMBER(38, 0) DEFAULT 0,
"M_TYPENAME5V0" VARCHAR2(1000),
"M_TABLENAME6V0" VARCHAR2(1000),
"M_STREXTERNALID7V0" VARCHAR2(1000),
"M_STROBJECTUUID8V3" VARCHAR2(1000),
"M_STRORIGINALUUID9V8" VARCHAR2(1000),
"M_STROEMOBJECTUUID10V13" VARCHAR2(1000),
"M_CREATIONDATE8V0" DATE,
"M_MODIFICATIONDATE9V0" DATE,
"M_BWRITECHANGEPROTOCOL10V0" CHAR(1) DEFAULT chr(0),
"M_DBLSORTINDEX12V0" NUMBER DEFAULT 0,
"M_STRCREATOR13V0" VARCHAR2(1000),
"M_STRMODIFIER14V0" VARCHAR2(1000),
"M_PUSEROWNERID20V8" NUMBER(38, 0) DEFAULT 0,
"M_PUSEROWNERID21V8" NUMBER(38, 0) DEFAULT 0,

```

```
"STRPLMID" VARCHAR2(1000),  
"BSIMPLESETEXIST" CHAR(1) DEFAULT chr(0),  
PRIMARY KEY ( OID ) )  
/
```