

SAP-Systeme mit externen Oracle-Datenbanken

SAP-Workload-Management für Oracle-DB

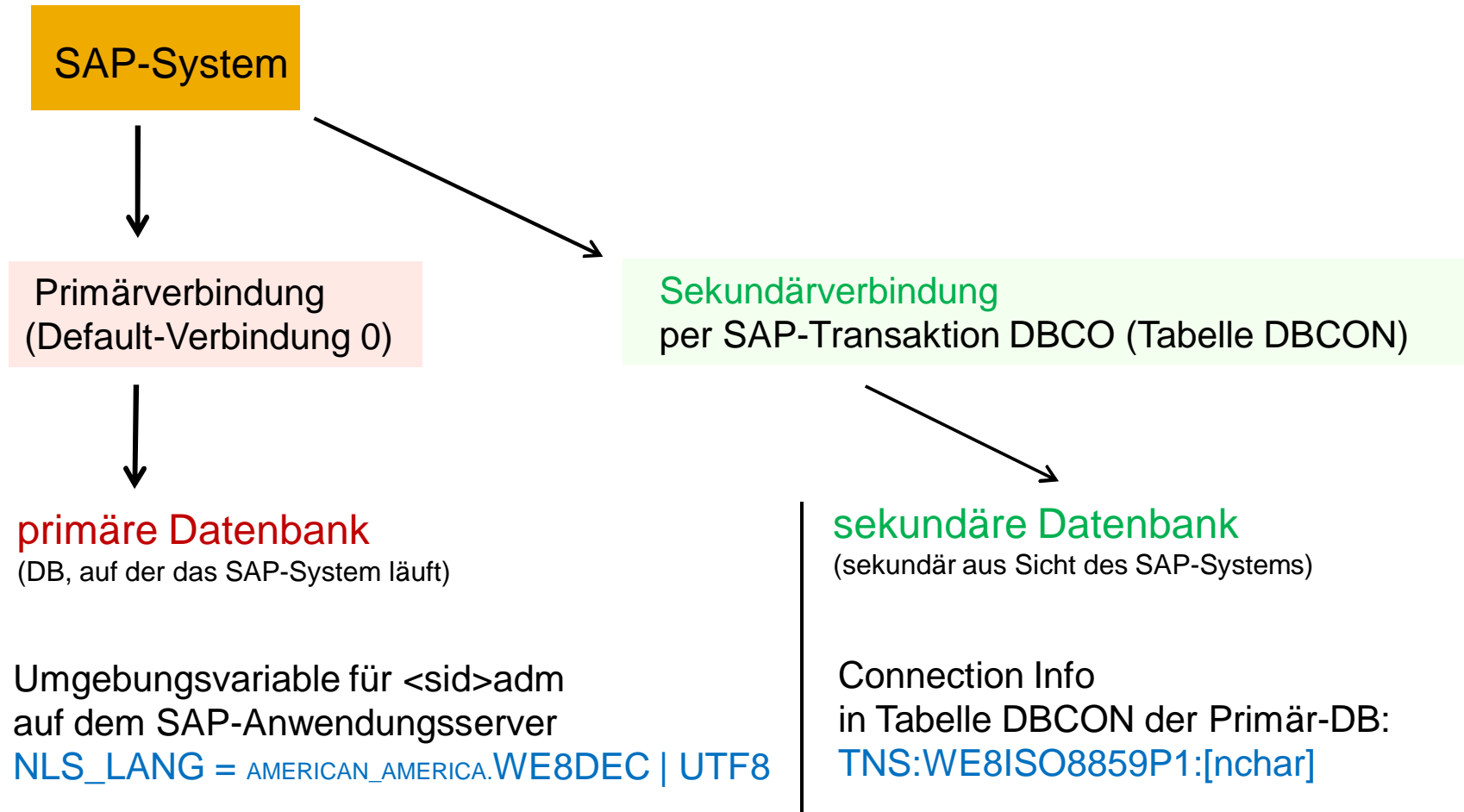


Manfred Riemer
SAP AG, Walldorf
Programmentwickler
Datenbank-Schnittstelle SAP/Oracle
16.11.2011

1. Sekundärverbindung zu externer Oracle-DB



Im ersten Teil des Vortrags geht es heute nachmittag um **Sekundärverbindungen** von SAP zu einer externen Oracle-DB (die **Primär-DB** muß keine Oracle-DB sein).



Die allgemeine Syntax für das Feld Connection Info in Transaktion DBCO ist:

tns:[charset]:[ncharset]

- tns: TNS-Alias der externen Oracle-DB (muß immer angegeben werden), an dieser Stelle kann auch ein sogenannter Easy-Connect-String stehen
- charset: Zeichensatz der externen DB, nur anzugeben, wenn Oracle-Konvertierung der VARCHAR2-Daten zur externen DB unterdrückt werden soll
- ncharset: analog zu charset, wird jedoch bisher nicht benutzt (außer Schlüsselwort NCHAR)

Den Wert ncharset (nationaler Characterset der remote DB) bitte niemals angeben, dieses Feld ist reserviert für eventuelle zukünftige Anwendungen.

Eine Ausnahme ist das Schlüsselwort NCHAR, siehe später.

Zur Zeit werden Inhalte von NCHAR- und NVARCHAR2-Feldern immer automatisch von Oracle konvertiert.

1. Sekundärverbindung



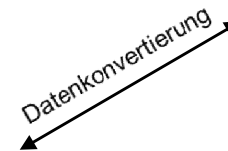
primäre SAP-Datenbank

```
V$NLS_PARAMETERS  
NLS_CHARACTERSET = WE8DEC | UTF8  
NLS_NCHAR_CHARACTERSET = UTF8 | UTF16
```



sekundäre Oracle-Datenbank

```
V$NLS_PARAMETERS  
NLS_CHARACTERSET = WE8ISO8859P1  
NLS_NCHAR_CHARACTERSET = UTF8 | UTF16
```



SAP-Anwendungsserver (WE8DEC | UTF16)

Umgebungsvariable für OS-User <sid>adm
NLS_LANG = AMERICAN_AMERICA.WE8DEC | UTF8

SAP-Transaktion **DBC0**, Feld Connection Info,
z.B.: **TNS:WE8ISO8859P1:[ncharset]**

Die **Datenkonvertierung erfolgt automatisch** von Oracle entsprechend dem DB-Zeichensatz (V\$NLS_PARAMETERS) und der Client-Umgebung (NLS_LANG).

Der sekundäre DB-Zeichensatz (hier charset=WE8ISO...) darf also im Normalfall **NICHT** in DBCO definiert werden!

Nur wenn die automatische Konvertierung **verhindert** werden soll, muß die Connect-Umgebung des SAP-Anwendungsservers vom Default NLS_LANG auf den Zeichensatz der Sekundär-DB umgestellt werden. Dazu wird der sekundäre Zeichensatz in DBCO definiert.

Schlüsselwort **NCHAR**

Ausnahme:

NCHAR bedeutet keinen Zeichensatz, sondern ist ein festes Schlüsselwort für das Feld [ncharset] in der Connect-Info der Transaktion DBCO.

Hierbei handelt es sich um eine Ausnahme zur aktuellen Regelung (siehe voriges Bild), daß das Feld [ncharset] nur für evtl. zukünftige Bedürfnisse vorgesehen ist und aktuell nicht benutzt wird.

Beispiel: **TNS::NCHAR**

Dieser Eintrag ermöglicht es, bei der Sekundärverbindung auf die Daten vom Typ NVARCHAR2 anstelle des Typs VARCHAR2 zuzugreifen.

Der Zeichensatz für NVARCHAR2 muß Unicode (z.B. UTF8 oder UTF16) sein.
Die Primärdatenbank kann nur eine Unicode-DB sein.

SAP-Hint `SAP_RAW_TO_CHAR`

wird nur für den Fall benötigt, daß gleichzeitig folgende 2 Voraussetzungen vorliegen:

1. Die Primär-DB hat Unicode-Zeichen und die Sekundär-DB hat einen 1-Byte-Zeichensatz.
2. Die Datenkonvertierung soll nicht automatisch von Oracle vorgenommen werden, sondern sie erfolgt, wenn notwendig, im Anwendungsprogramm (z.B. mit den ABAP-Funktionen `CL_ABAP_CONV_IN_CE`, `CL_ABAP_CONV_OUT_CE` o.a.)

Der Hint ermöglicht

- das Einfügen (INSERT) von Raw-Daten (einzelnen Bytes) von ABAP-Raw-Typen (XSTRING) in Characterfelder (VARCHAR2, nonunicode) der Sekundär-DB oder
- das Lesen (SELECT) von Raw-Daten (Bytes) aus Characterfeldern (VARCHAR2, nonunicode) der Sekundär-DB in den ABAP-Raw-Typ XSTRING.

Der Hint ist nur beim ABAP-Typ XSTRING erlaubt, d.h. das Statement darf keine Variablen von Typ X, Typ C oder Typ STRING enthalten!

Wenn ein Statement diesen Hint enthält, dann werden ABAP-Raw-Daten (Bytes) ohne eine Konvertierung seitens Oracle in Tabellen-Characterfelder (VARCHAR2) eingefügt bzw. aus Characterfeldern der Tabelle unkonvertierte Bytes in ABAP-Raw-Felder gelesen.

1. Sekundärverbindung



Beispiel für den Pseudo-Hint **SAP_RAW_TO_CHAR** im ABAP-Programm

in Native-SQL-Statements an einer beliebigen Stelle als einfacher Kommentar:

```
EXEC SQL PERFORMING load_wa_xstr.  
    SELECT          itsid  
    INTO           :wa_xstr-itsid  
    FROM           survey_feedback_master  
    /* SAP_RAW_TO_CHAR */  
ENDEXEC.
```

oder

```
EXEC SQL.  
    INSERT /* SAP_RAW_TO_CHAR */  
    INTO       EM.VND(VENDOR,LANG,TEXT) VALUES (:A0, :A1, :A2 )  
ENDEXEC.
```

oder im ABAP-Statement (Open SQL) am Ende des Statements in der Form:

```
%_hints ORACLE 'SAP_RAW_TO_CHAR'
```

- Er wird nur von der SAP-Datenbankschnittstelle (DBSL) ausgewertet, nicht an Oracle weitergereicht.
- Wenn es sich nicht um ein Unicode-SAP-System und eine Sekundärverbindung zu einer DB mit 1-Byte-Zeichensatz (WE8DEC u.a., auch US7ASCII) handelt, dann wird der Hint ignoriert.
- Der Hint kann nur in Verbindung mit dem ABAP-Typ XSTRING benutzt werden, andernfalls bricht das Programm mit einem Kurz-Dump und entsprechender Fehlermeldung im Workprozeß-Protokoll ab.

2. SAP-Workload-Management für Oracle-DB



Der zweite Vortragsteil befaßt sich mit dem Workload-Management unter Nutzung des Oracle-Resource-Managers.

(Hierbei geht es im Gegensatz zum ersten Teil um SAP-Systeme mit einer primären Oracle-DB.)

Das SAP-Workload-Management beruht auf der sogenannten **Application Info**, die in den Oracle-Systemtabellen abgespeichert wird und nutzt die Oracle-Funktionalität des Resource-Managers. Indem bestimmte Arbeiten bestimmten SAP-Usern zugeordnet werden, können diese in der Oracle-Session verschieden priorisiert werden.

SAP-Profile-Parameter

db/ora/register_appl_info = 1 | 0 **Default: 1**

Bei 0 wird die Ermittlung und Speicherung der Application Info unterdrückt. Das ist nützlich fürs Tracing/Debugging oder wenn die Oracle-DB nur extern (über Sekundärverbindungen) benutzt wird. Für das SAP-Workload-Management muß daher immer die Defaulteinstellung beibehalten werden, d.h. der Parameter kann entweder entfallen oder darf zumindest nicht explizit auf 0 gesetzt sein.

2. SAP-Workload-Management für Oracle-DB



ABAP-Laufzeit-Umgebung (Application Info)

<u>Name</u>	<u>Inhalt</u>
module	ABAP-Programm
action (position)	Ort innerhalb des ABAP-Programms (Zahl, aus der die ABAP-Zeilennr. berechnet wird)
client (user)	SAP-Benutzername
transaction	SAP-Transaktionsname
mainprogram	ABAP-Hauptprogramm
connection	DBSL-interne Verbindungsnummer (Connection-Handle) zur Datenbank

Die DB-Schnittstelle (dboraslib.<ext>) speichert diese Information in 2 Oracle-Systemtabellen ab:

Oracle-Datenbank



V\$SESSION (Bytes)

MODULE (48)
ACTION (32)
CLIENT_IDENTIFIER (64)
CLIENT_INFO (64)

V\$SQL (Bytes)

MODULE (64)
ACTION (64)

2. SAP-Workload-Management für Oracle-DB



V\$SESSION

Feldinhalt ("Application Info")

MODULE:	module	(Name des ABAP-Moduls)
ACTION:	position	(Ort des Statements im Modul)
CLIENT_IDENTIFIER:	user	(ausführender SAP-User)
CLIENT_INFO:	connection:transaction:mainprogram	(Erklärung siehe voriges Bild)

Die ersten beiden Spalten (MODULE und ACTION) sind mit gleichem Inhalt auch in Tabelle V\$SQL enthalten.

Aktualisierung der Application Info:

- vor jeder ersten Ausführung (PREPARE) eines SQL-Statements:

Alle Angaben werden bestimmt.

- bei jeder weiteren Ausführung (EXECUTE) eines SQL-Statements innerhalb der gleichen Transaktion:

Das Feld CLIENT_IDENTIFIER wird mit dem Wert von **user** aktualisiert, die anderen Felder bleiben unverändert.

Praktisches Beispiel:

- Alle SAP-Dialog-Transaktionen von User **SAPUSER** sollen unter den Bedingungen der Consumer-Gruppe **SYS_GROUP** laufen.
- Schritte:
 1. SQL> alter system set **resource_manager_plan** = 'DEFAULT_PLAN' scope = both;
 2. SQL> exec **dbms_resource_manager.clear_pending_area** ();
 3. SQL> exec **dbms_resource_manager.create_pending_area** ();
 4. SQL> exec **dbms_resource_manager.set_consumer_group_mapping** (
attribute => **dbms_resource_manager.client_id**,
value => **'SAPUSER'**,
consumer_group => **'SYS_GROUP'**);
 5. SQL> exec **dbms_resource_manager.submit_pending_area** ();
- Das Feld **CLIENT_IDENTIFIER** der Tabelle **V\$SESSION** enthält das Group-Mapping: den ABAP-Usernamen.
- Wenn eine SAP-Transaktion unter dem Dialog-User SAPUSER läuft, wechselt die korrespondierende Oracle-Session des SAP-Work-Prozesses automatisch zur Resource-Gruppe **SYS_GROUP**. Wird ein späteres Statement von einem anderen, nicht privilegierten SAP-User ausgeführt, schaltet die Oracle-Session automatisch zurück zur Default-Resource-Gruppe **OTHER_GROUPS**.

Schluß -

vielen Dank für Ihr Interesse!