

Betrieb eines Apex-Servers: Was der DBA beachten sollte

Carsten Czarski, ORACLE Deutschland B.V. & Co. KG

Mit zunehmender Anzahl von Applikationen rückt Application Express (Apex) mehr und mehr in den Fokus der Datenbankadministration. Da Apex komplett in der Oracle Datenbank läuft, können viele vorhandene Betriebskonzepte wie Backup und Recovery oder die Konfiguration von Standby-Datenbanken „1:1“ auf eine Apex-Installation angewendet werden. Allerdings hat Apex auch einige Besonderheiten, die beim Betrieb zu beachten sind. So kommen mit dem Einrichten und Verwalten der Workspaces oder der Überwachung von Apex-Anwendungen auch zusätzliche Aufgaben auf den DBA zu.

Apex ist von Haus aus mandantenfähig; Anwendungen werden in Workspaces, autonomen Bereichen mit eigener Benutzerverwaltung, entwickelt. Eine Apex-Installation in einer Datenbank kann beliebig viele Workspaces enthalten. Apex bringt somit Workspace-bezogene und zentrale Administrationsaufgaben mit. Die zentralen Aufgaben werden am besten vom DBA übernommen, da sie direkte Auswirkungen auf den Datenbankbetrieb haben. So kann das Erstellen eines neuen Workspaces die Einrichtung eines Datenbankschemas (CREATE USER) zur Folge haben. Die Verwaltung des Workspaces selbst, also dessen Nutzer oder Applikationen,

findet dagegen dezentral durch den Eigentümer statt; der DBA muss hier normalerweise nichts tun.

Werkzeuge zur Administration eines Apex-Servers

An erster Stelle steht der System-Workspace „INTERNAL“ (siehe Abbildung 1). Hier führt der DBA alle zentralen Verwaltungsaufgaben per Web-Browser durch. Der Workspace INTERNAL unterscheidet sich signifikant von einem normalen Apex-Workspace und stellt folgende Bereiche zur Verfügung:

- Workspace-Verwaltung

- Verwaltung von Apex-Instanzeinstellungen
- Überwachung

Zentral eingestellt wird beispielsweise der Mailserver für Apex_MAIL, das von Apex eigens bereitgestellte PL/SQL-Paket zum Mailversand. Auch die PDF-Engine (Oracle BI Publisher oder Apache FOP), die für die Aufbereitung von Berichten verwendet wird, ist im Workspace INTERNAL konfiguriert. Dazu kommen allgemeine Richtlinien für alle Apex-Workspaces wie Passwort-Richtlinien oder die maximale Dauer einer Apex-Benutzersitzung. Neben der Web-Oberfläche kann der DBA

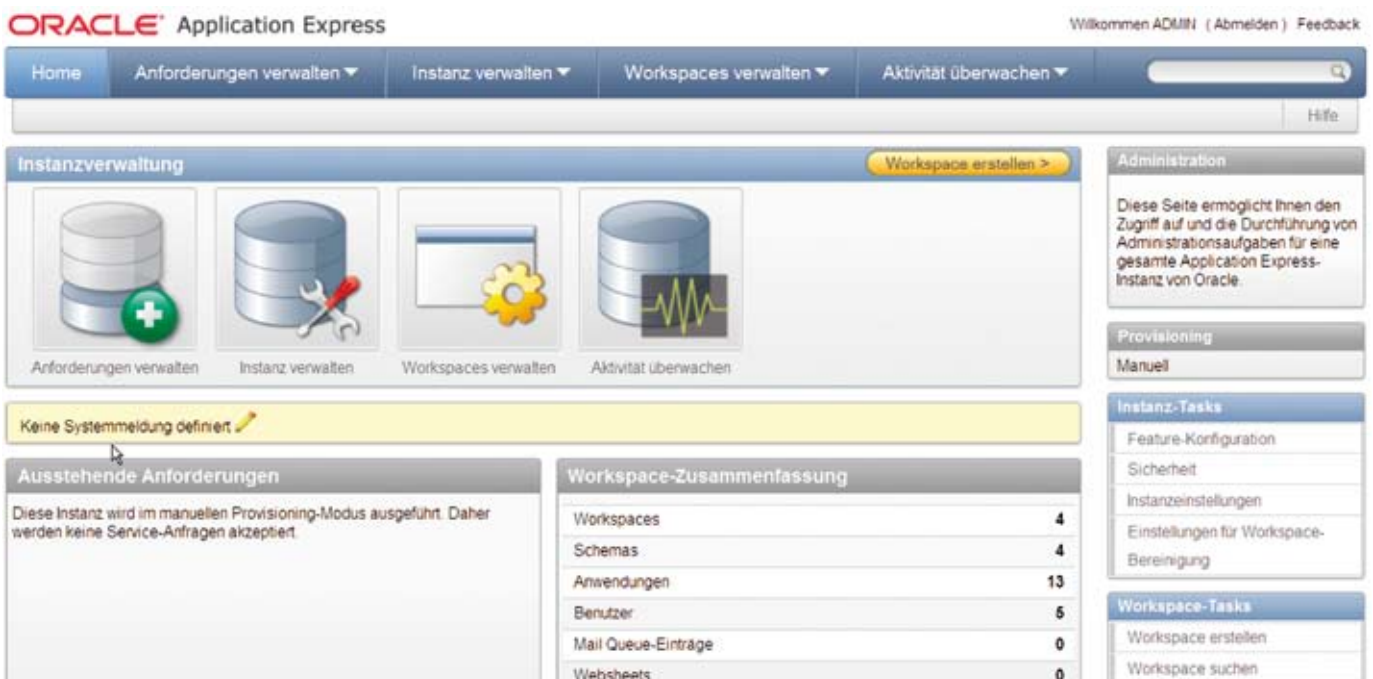


Abbildung1: Zentraler Administrationsbereich (Workspace INTERNAL)

allerdings auch mit dem PL/SQL-Paket `Apex_INSTANCE_ADMIN` arbeiten. Listing 1 zeigt, wie man den Mailserver auch ohne Browser einstellen kann.

Analog dazu stellt `Apex_INSTANCE_ADMIN` auch Prozeduren zum Anlegen oder Bearbeiten von Workspaces bereit. Abbildung 2 zeigt eine Erweiterung für den Oracle SQL Developer, die auf `Apex_INSTANCE_ADMIN` basiert und frei heruntergeladen werden kann (siehe „Weitere Informationen“).

Apex-Installation: Die Wahl des Webservers

Nachdem die erste Apex-Version nur mit dem auf dem Apache basierenden Oracle HTTP Server als Webserver betrieben werden konnte, stehen nun zwei weitere Alternativen bereit:

- Embedded PL/SQL Gateway der Datenbank
- Apex J2EE Listener

Das Embedded PL/SQL Gateway ist vor allem für Entwicklersysteme geeignet; im Fokus steht eine möglichst einfache Apex-Installation, die – außer Datenbank und Listener – keine weiteren Komponenten mehr braucht. Das Embedded Gateway basiert auf der Shared-Server-Architektur der Datenbank; Einschränkungen bestehen bei den Webserver-Features: So stehen „URL Rewriting oder Proxyserver“-Funktionen nicht zur Verfügung. Allerdings ist das Embedded Gateway von Haus aus nicht besonders gut konfiguriert. Es empfiehlt sich, die Werte der Parameter `SHARED_SERVERS` und `MAX_SHARED_SERVERS` zu erhöhen. Die Apex-Community (<http://www.oracle.com/global/de/community/index.html>) liefert nähere Informationen.

Der Apex J2EE-Listener basiert komplett auf der Java-Technologie, wird von Oracle als EAR-Archiv bereitgestellt und lässt sich prinzipiell in jeden J2EE-Applikationsserver installieren. Der J2EE-Listener bietet sich vor allem dann an, wenn Oracles Apache-Server, der auf einer ganz bestimmten Apache-Version basiert, nicht in den Rechenzentrum-Betrieb passt.

Apex Workspaces

Bereits bei der Konzeption des Apex-Servers stellt sich die Frage, wie die Workspaces den Entwicklern zur Verfügung stehen sollen. Apex kennt verschiedene Bereitstellungsvarianten – im Workspace `INTERNAL` wird in den Instanz-Einstellungen festgelegt, welche aktiv ist (siehe Abbildung 3).

Standardmäßig ist „Manuell“ eingestellt, das bedeutet, dass neue Workspaces nur vom DBA per Web-Oberfläche oder mit `Apex_INSTANCE_ADMIN` bereitgestellt werden. Beim Umstellen auf „Anfordern“ erhält die Apex-Login-Seite einen zusätzlichen Link „Workspace anfordern“. Hier kann der Interessent in Selbstbedienung einen Workspace beantragen, der dann vom DBA nur noch genehmigt oder abgelehnt werden muss. Die dritte Variante „E-Mail-Verifizierung“ geht noch darüber hinaus: Hier sendet Apex, nachdem der Workspace beantragt wurde, eine Mail an die angegebene Adresse. Wird diese dann, wie im Internet üblich, mit Klick auf einen speziellen

```
begin
apex_instance_admin.set_parameter(
,SMTP_HOST_ADDRESS', ,mailserver.de'
);
apex_instance_admin.set_parameter(
,SMTP_HOST_PORT', ,25'
);
end;
/
commit
/
```

Listing 1: Einstellen des Apex-Mailservers mit PL/SQL in SQL*Plus

Link bestätigt, richtet Apex den Workspace automatisch ein.

Welche Variante die richtige ist, hängt vom Einsatz des Apex-Servers im Unternehmen ab. Dient er als Laufzeit- oder Entwicklungsumgebung für dedizierte Apex-Anwendungen, so wird man am ehesten mit manueller Bereitstellung arbeiten, denn in diesen Fällen wird der DBA die Einrichtung von Workspaces selbst vornehmen wollen.

Ein Apex-Server kann aber auch, ganz nach dem Vorbild von `apex.oracle`.

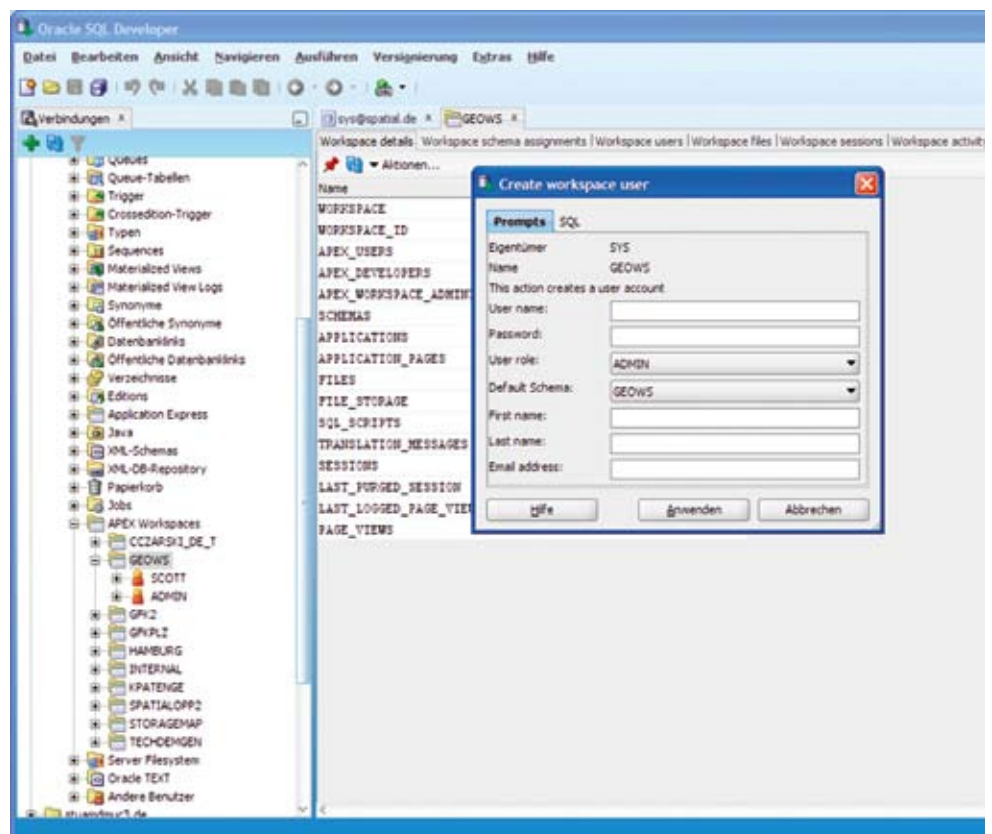


Abbildung 2: SQL Developer Plug-in zu Apex Workspace Verwaltung

com als „öffentlicher“ Server im Unternehmen betrieben werden. Bei diesem Betriebsmodell kann jeder interessierte Mitarbeiter einen Workspace beantragen und danach mit der Entwicklung von Apex-Anwendungen loslegen. Dies ist ein interessanter Ansatz, die taktische Anwendungsentwicklung von den Abteilungs-PCs auf die zentral betriebene Apex-Instanz zu holen.

Wichtig bei einem solchen „öffentlichen“ Apex-Server ist der Umgang mit inaktiven Workspaces. Wie bei apex.oracle.com melden sich die meisten Nutzer nicht formal ab, wenn der Workspace nicht mehr gebraucht wird. Apex 4.0 bringt daher den Prozess zum „Aufräumen“ inaktiver Workspaces mit, den die Nutzer von apex.oracle.com schon kennen. Abbildung 3 zeigt den Dialog, in dem unter anderem festgelegt wird, wie lange ein Workspace inaktiv sein muss, damit er zum Löschen vorgemerkt wird oder wie viele Tage im Voraus der Workspace-Eigentümer benachrichtigt werden soll. Reagiert der Nutzer auf die Benachrichtigung nicht, so wird der Workspace nach Ablauf der Frist gelöscht. Klickt er dagegen auf den in der E-Mail enthaltenen Link („do not purge Workspace“), so wird der Workspace beibehalten.

Entwicklung, Test und Produktion

Wird ein Apex-Server für dedizierte Anwendungen eingerichtet, so stellt sich häufig die Frage nach Entwicklungs-, Test- und Produktionssystemen. Die Trennung ist auch für Apex-Systeme sinnvoll, besonders dann, wenn die Weiterentwicklung der Anwendung und das Bereitstellen neuer Versionen einem kontrollierten Prozess folgen sollen.

Für das Produktiv- und auch für das Testsystem kann der DBA die Apex-Entwicklungsumgebung mit dem Skript „apxdevrm.sql“ (im Apex-Home-Verzeichnis) entfernen. Damit ist automatisch sichergestellt, dass keine Änderungen an der produktiv eingesetzten Anwendung stattfinden und der Prozess von Entwicklung, Abnahme und Produktion stets eingehalten wird. Allerdings sollte man sich das gut überlegen, denn das schnelle und unkomplizierte Ändern einer Anwendung – ohne jeden Deployment-Prozess – ist eine Stärke von Apex, die für eilige Bugfixes sehr wertvoll sein kann.

Ist die Entwicklungsumgebung entfernt, so gilt das auch für den Workspace INTERNAL; die Administration erfolgt dann allein mit dem PL/SQL-Paket Apex_INSTANCE_ADMIN. Die Übertragung einer Anwendung vom

Entwicklungs- auf das Test- beziehungsweise Produktionssystem erfolgt per Ex- und Import. Letzterer findet mit SQL*Plus statt, denn ein Apex-Anwendungsexport ist immer ein SQL-Skript. Ad-hoc-Änderungen an Apex-Anwendungen sind auf einem solchen System nicht mehr möglich.

Wenn die Workspaces auf den verschiedenen Systemen auf normalem Weg vom DBA angelegt werden, erhalten sie unterschiedliche (interne) IDs. Das führt jedoch beim Einspielen der Exportdatei mit SQL*Plus zu einer Fehlermeldung, da diese die Workspace-ID des Quellsystems verwendet. Das direkte Ändern der Exportdatei ist in der Theorie möglich, aber nicht zu empfehlen. Ab Apex 4.0 lassen sich Workspace- und Anwendungs-ID vor dem Einspielen der Exportdatei mit dem Paket Apex_APPLICATION_INSTALL setzen. Voraussetzung ist aber, dass die Anwendung aus einer Apex 4.0-Umgebung exportiert wurde.

Gleiche Workspace-IDs können sichergestellt werden, wenn der Workspace auf einem System erstellt und danach per Workspace-Export und -Import auf die anderen Systeme übertragen wird. In Apex 4.0 ist es möglich, die ID beim Erstellen des Workspaces explizit anzugeben.

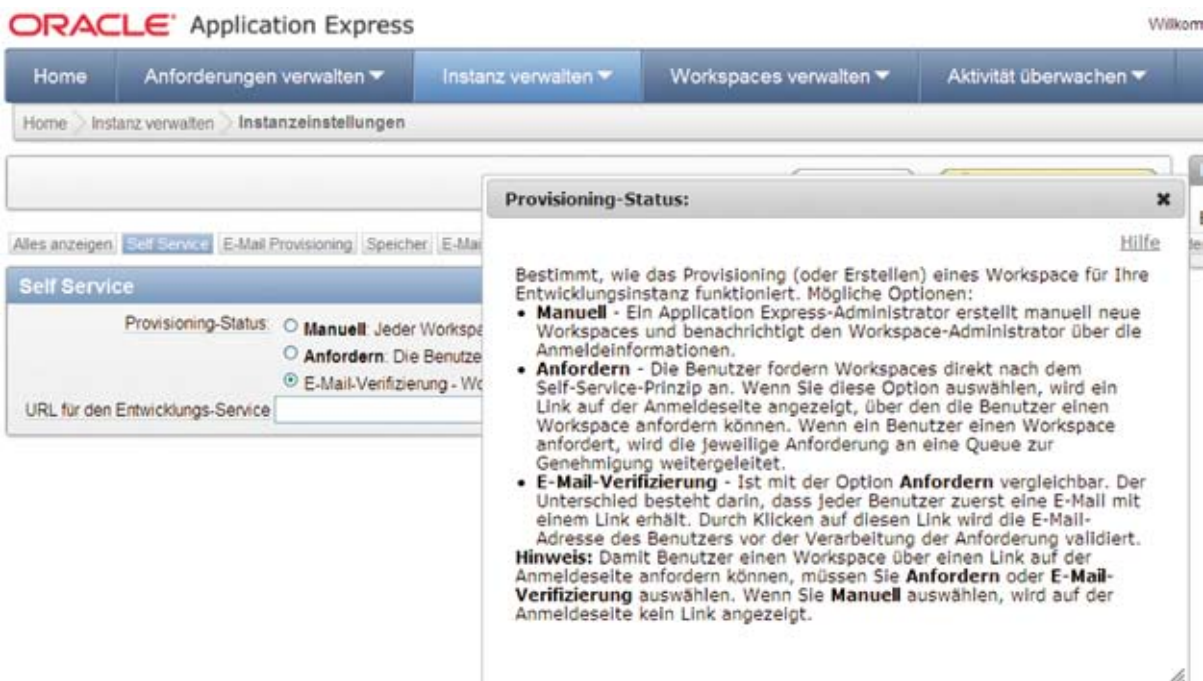


Abbildung 3: Einstellen des Bereitstellungsmodus für neue Workspaces

Apex-Anwendungen

Auch jede Anwendung erhält in Apex eine ID. Wird eine Applikation nun per Ex- und Import von einem System auf ein anderes übertragen, kann man die gleiche ID wiederverwenden, falls sie auf dem Zielsystem noch frei ist. Ist sie dagegen belegt, ist beim Import eine andere zu wählen – das kann in Apex 4.0 auch in SQL*Plus mit Apex_APPLICATION_INSTALL geschehen. Für Apex-Entwickler gibt es die Umgebungsvariable APP_ID, sodass man die Applikations-ID etwa für Links nicht hart codieren muss. Eine einmal festgelegte Anwendungs-ID sollte sich dennoch auf dem Produktionssystem nach Möglichkeit nicht mehr ändern. Zwar löst eine geänderte Anwendungs-ID bei sauber entwickelten Anwendungen keine Fehlermeldungen aus, es gibt jedoch andere Gründe:

- Endanwender speichern Apex-URLs häufig als Bookmark ab – die Anwendungs-ID ist in der URL enthalten
- Übersetzungen sind in Exportdateien nicht enthalten; die XLIFF-Dateien werden nach dem Import erneut eingespielt. Das erfordert allerdings die gleiche Anwendungs-ID
- Wird die Performance der Anwendung überwacht, ist dies für die neue Anwendungs-ID anzupassen

Für Apex-Systeme mit dedizierten Anwendungen empfiehlt sich daher die Verwaltung der Anwendungs-IDs durch den DBA: Eine neue Applikation bekommt ihre ID vor Übernahme ins Test- oder Produktionssystem und behält diese dauerhaft.

Monitoring einer Apex-Installation

Zum Überwachen der Apex-Instanz stehen dem DBA eine Vielzahl von Apex Dictionary Views (siehe „Weitere Informationen“) zur Verfügung. Mit am wichtigsten ist das „Apex Activity Log“, in dem jeder Apex-Seitenabruf samt Detailinformationen und Antwortzeit mitgeschrieben wird. Möchte der DBA beispielsweise wissen, ob in den letzten 15 Minuten Antwortzeiten von mehr als fünf Sekunden aufgetre-

```
select
  flow_id application_id,
  step_id page_id,
  user_id user_id,
  elap    elapsed_time
from apex_040000.wv_flow_activity_log
where elap > 5
and time_stamp between sysdate - interval ,15' minute and sysdate
/
```

Listing 2: Abrufen aller Apex-Seitenabrufe der letzten 15 Minuten mit mehr als 5 Sekunden Antwortzeit

```
select username, action, module, client_info
from v$session
where username is not null order by 1;
```

USERNAME	ACTION	MODULE	CLIENT_INFO
Apex_PUBLIC_USER	PAGE 4150	Apex:APPLICATION 4000	ADMIN
GEOWS			JDBC Thin Client
GEOWS			JDBC Thin Client
MVDEMO			9 JDBC Thin Client
MVDEMO			JDBC Thin Client

Listing 3: Blick auf V\$SESSION mit Apex-Aktivität

ten sind, findet er das mit einer SQL-Abfrage heraus (siehe Listing 2).

Apex-Aktivität ist auch in anderen Performance-Views der Datenbank erkennbar – DBMS_APPLICATION_INFO wird konsequent verwendet. Listing 3 zeigt einen Blick auf die View V\$SESSION, während eine Datenbanksitzung mit einem Apex-Seitenabruf beschäftigt ist. Problematische SQL-Anweisungen lassen sich bei Bedarf also sofort einer Apex-Anwendung zuordnen – die Informationen sind in den Spalten MODULE und ACTION enthalten. Auf dieser Grundlage lassen sich Apex-Installationen mit Standardwerkzeugen wie Grid Control überwachen. Zwar sind out-of-the-box keine Apex-Metriken vorhanden, sie lassen sich aber einfach hinzufügen: Bei Grid Control wird eine „benutzerdefinierte Metrik“ auf Basis einer SQL-Abfrage eingerichtet – auf gleichem Wege kann Apex prinzipiell in alle Werkzeuge integriert werden. Das Abrufen der Metrik mit SQL ist alles, was zu tun ist.

Fazit

Übernimmt man als DBA die Betreuung einer Apex-Installation, so kann

man diese zunächst wie jede andere Datenbank betreiben. Allerdings sind Datenbank und Application Server in Apex vereinigt – es kommen also Aufgaben dazu. Der Mehraufwand hält sich für den geübten DBA, der die Tatsache, dass Apex in der Datenbank läuft, konsequent nutzt, allerdings in Grenzen.

Weitere Informationen

Auf den Webseiten von Oracle und im Internet ist umfangreiches Material auch in deutscher Sprache vorhanden:

1. Deutschsprachige Apex Community: <http://www.oracle.com/global/de/community/index.html>
2. Dictionary Views: <http://apex.oracle.com/url/apxrepo>
3. Vorfahrtsregeln: <http://apex.oracle.com/url/resman>
4. SQL Developer Plugin: <http://apex.oracle.com/url/apxsqldev>
5. EM Monitoring: <http://apex.oracle.com/url/apxem>
6. PL/SQL Embedded Gateway: <http://apex.oracle.com/url/websrv>

Kontakt:

Carsten Czarski
carsten.czarski@oracle.com