

Rollout Prozess für APEX Anwendungen

Oliver Lemm
MT-AG
Ratingen

Schlüsselworte:

APEX, Rollout, Development

Einleitung

Der Vortrag soll aufzeigen wie man eine APEX Anwendung für Installation und Upgrades optimal vorbereitet und durchführt. Es werden Probleme und Besonderheiten von APEX hervorgehoben und generelle Eigenschaften von Rollout Prozessen aufgegriffen.

Ziel des Rollout-Prozesses

Ein Rollout-Prozess schließt die Planung und Durchführung eines Updates oder die Installation einer Anwendung mit ein. Es muss sichergestellt werden, dass beim Ausrollen eine vollständige und korrekte Installation auf allen erforderlichen Systemen erfolgt. Eventuell auftretende Fehler sollten in einem Protokoll festgehalten werden. Es muss sichergestellt werden, dass ein Zugriff durch die Benutzer während der Installation nicht möglich ist.

Bei einem Update müssen zusätzliche Aspekte berücksichtigt werden, wie die Absicherung der alten Anwendung und Daten und dass ein Update nur dann eingespielt werden darf, wenn eine gültige Vorversion vorhanden ist. Weiter dürfen bei eventuell nötigen Datenmodelländerungen keine alten Daten verloren gehen, sondern müssen korrekt migriert werden.

Was deckt APEX ab

APEX als Framework auf der Datenbank besitzt einerseits alle Vorteile der Datenbank, welche innerhalb eines Rollout Prozesses nur Änderungen am Datenmodell zulässt die auch durchführbar (kompilierbar) sind. Darüber hinaus lassen sich beliebige PL/SQL oder SQL Scripts benutzen, welche man schnell und einfach in Testumgebungen vorbereiten kann. Andererseits besitzt es die Vorteile eines Frameworks, das im Falle von APEX ein sehr schneller browserbasierter Entwicklungsbereich ist, der keinerlei weitere Werkzeuge bei der Installation voraussetzt.

Bevor man eine Installation oder ein Update durchführt kann man die APEX-Anwendung mit einer Mitteilung sperren. Diese Einstellung findet man unter „Anwendungseigenschaften“ bei „Verfügbarkeit“. So ist gewährleistet, dass kein Zugriff mehr auf die Anwendung möglich ist und der Benutzer über die Meldung genauere Infos bekommen kann. Da der Status beim Export einer Anwendung mitgespeichert wird, empfiehlt es sich, diesen vor dem Export (in der Entwicklungs-/Testumgebung) auf „nicht verfügbar“ zu stellen, um nach der Installation zunächst Tests und Prüfungen innerhalb der Datenbank durchführen zu können, ohne dass ein Anwender die Anwendung ausführen kann.

Alles anzeigen	Name	Eigenschaften	Verfügbarkeit	Globale Benachrichtigung	Ersetzungen	Logo	Theme	Template-Standards	Komponentenstandards
Verfügbarkeit									
Status	Nichtverfügbar								
Build-Status	Anwendung nur ausführen								
Nichtverfügbarkeitsmeldung	Die Anwendung ist wegen eines angekündigten Updates nicht verfügbar.								
Auf durch Kommas getrennte Benutzerliste einschränken (Status muss "Eingeschränkter Zugriff" lauten)									

Abbildung 1: Verfügbarkeit der Anwendung einstellen

Es empfiehlt sich natürlich vor jeder Installation oder jedem Update eine Sicherung der bisherigen Produktivdaten und der Anwendung vorzunehmen, welche über einen Datenexport und einen Export der Datenbankobjekte durchgeführt werden kann.

Bei einer Installation/Update wird in APEX immer zuerst die Anwendung/Oberfläche ersetzt und danach werden die Änderungen am Datenmodell und ggf. den Daten vorgenommen. Da die Anwendung selber eine standardkonforme PL/SQL Datei ist, kann sie sowohl über SQL Plus installiert werden, als auch über den Workspace mittels eines Browsers. Dabei enthält diese Datei die entsprechende APEX Version, in der die Anwendung entwickelt wurde um eine Installation nur in einer gleichen oder neueren Version zuzulassen.

APEX bietet die Möglichkeit sogenannte „Unterstützende Objekte“ direkt in der Anwendung zu hinterlegen. Mit diesen kann man sowohl das Datenmodell als auch die Daten der Anwendung aktualisieren. Diesen Bereich findet man innerhalb des Application Builder bei den Gemeinsamen Komponenten jeder APEX-Anwendung.

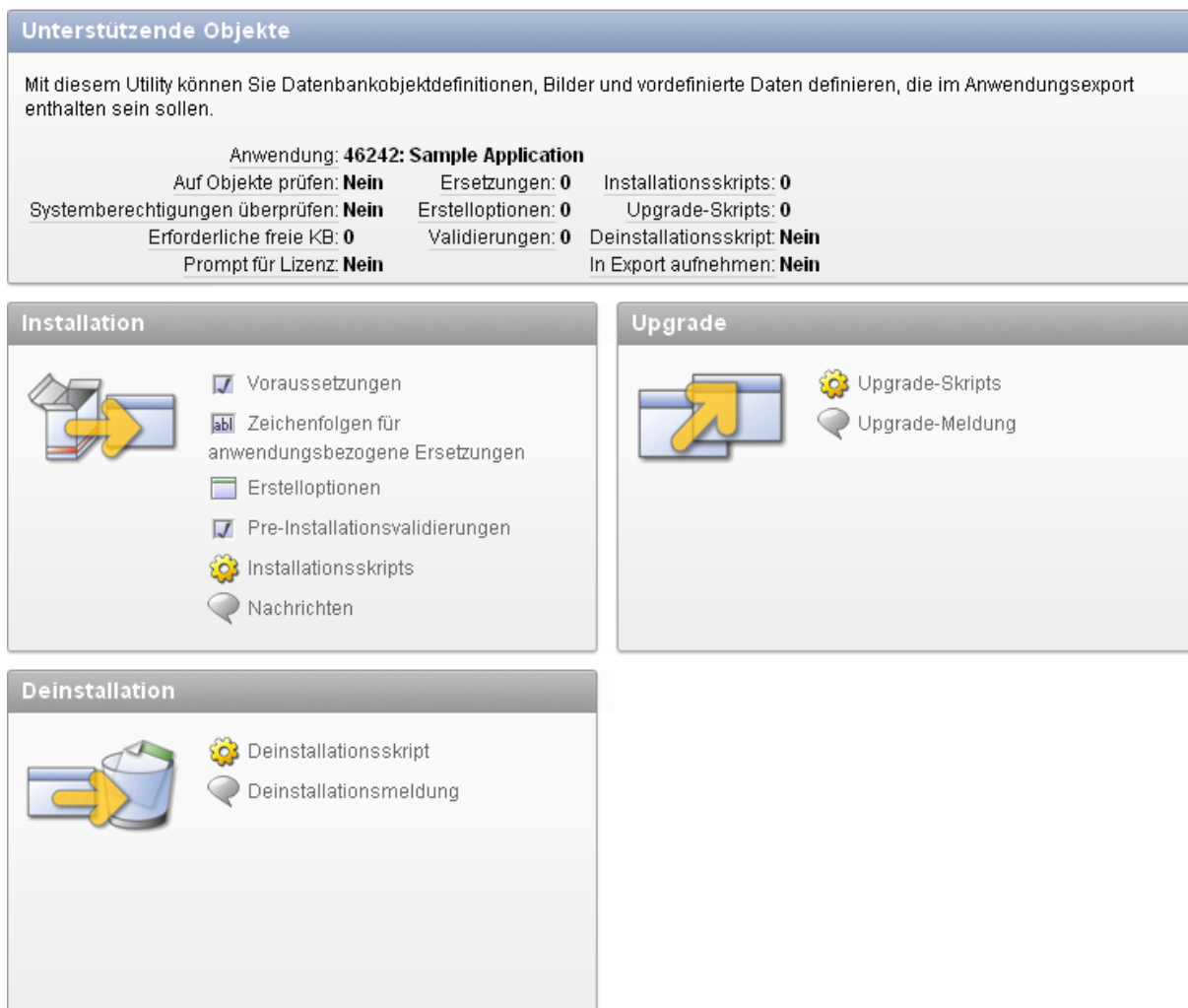


Abbildung 2: Unterstützende Objekte in APEX

Für die Installation kann man alle nötigen Meldungen hinterlegen, die zwischen den einzelnen Schritten einer Installation oder einem Updates ausgegeben werden. So kann der Benutzer einfach und schnell durch die Schritte der Installation geführt werden. Auch wenn eine Installation/Update einer APEX Anwendung meistens von Fachpersonal/DBA's durchgeführt wird können diese Nachrichten hilfreiche Hinweise über die durchzuführenden Änderungen enthalten. Weiter kann man sowohl eine allgemeine Validierung, als auch jeweils einzelne Validierungen vor zusätzlichen Skripten durchführen. Dies stellt sicher, dass nur dann eine Installation oder ein Update der entsprechenden Objekte durchgeführt wird, wenn bestimmte Rahmenbedingen erfüllt sind.

Desweiteren kann man auch ein Deinstallationskript mit angeben, damit eine Anwendung beim Löschen alle zugehörigen Objekte wieder aus der Datenbank entfernt. So bleiben keine Objekte in der Datenbank zurück und man muss nicht nach einer Entfernung der Anwendung von Hand entsprechende Objekte ausfindig machen und löschen.

Neben den SQL Skripten bietet APEX auch die Möglichkeit Bilder, CSS-Dateien, Themes und weitere statische Dateien separat zu ex- bzw. importieren. So kann man auch solche Objekte einzeln ex- und importieren.

Zuletzt wird am Ende einer jeden Installation/Update eine Zusammenfassung aller Änderungen angezeigt und ob diese erfolgreich waren.

Welche Probleme gibt es

Trotz der vielen Vorteile und Absicherungen gibt es immer wieder verschiedene Möglichkeiten warum eine Installation/Update nicht korrekt und sauber durchläuft. Dies kann an verschiedensten Einstellungen oder Beschränkungen der Datenbank oder des Datenmodells liegen. Läuft ein DDL Script einmal sind die Änderungen nicht mehr so einfach zurückzuführen, daher sollte man mindestens eine erfolgreiche Testinstallation durchführen.

Außerdem kann bei sehr vielen der Änderungen die Anzahl der Skripte zum Installieren oder Updaten sehr groß sein, so dass es sehr aufwendig ist die Skripte alle einzeln in APEX zu hinterlegen. In diesem Falle sollte man eigene Installationsskripte über SQL Plus einspielen, die über eine Logdatei (mittels spool on / spool off) und möglichst viele Kommentare (mittels prompt) eine möglichst genaue Übersicht bieten.

Ein weiteres Problem bei vielen einzelnen Versionen einer Anwendung ist, dass es sehr viele Änderungen am Datenmodell geben kann. Daher ist es wichtig das Datenmodell möglichst umfassend vorher zu planen um nicht immer wieder Datenmigrationen durchführen zu müssen. Man sollte innerhalb der Anwendung mit Validierungen vor Updateskripten sicherstellen, dass durchzuführende Änderungen auch beim momentanen Stand der Anwendung passen und dies zum Beispiel mittels einer Version zu gewährleisten.

Beim Import von CSS-Dateien hat APEX momentan noch das Problem, dass Dateien die in einer früheren Version installiert wurden, dort nicht überschrieben werden können. Hinterlegt man solche Dateien bei den Bildern der Anwendung ist es dennoch möglich diese in APEX zu speichern und bedingt durch den gleichen Pfad der Bilder entstehen keinerlei weitere Unterschiede. Die beiden Bereiche der CSS-Dateien und Bilder unterscheiden sich also nur zur Übersicht voneinander. Generell ist das Problem bei Bildern, CSS-Dateien und anderen statischen Dateien, dass diese bei Hinterlegung innerhalb der APEX-Anwendung nicht gecached werden. Man sollte also bei großen Dateien oder Systemen die auf Performance optimiert werden, die Dateien nicht innerhalb APEX hinterlegen, sondern im Dateisystem des Servers speichern und durch einen Apache Webserver cachen.

Leider bietet APEX neben der automatischen Validierung der Installationsdatei auf die APEX-Version keinerlei eigens definierte Möglichkeiten die Installation der Anwendung zu validieren. Nimmt man also aus Versehen eine Installationsdatei einer früheren Version der Anwendung und installiert diese, so wird sie ohne Fehler installiert und frühestens bei der Validierung der unterstützenden Objekte wird ein Fehler bei der dortigen Überprüfung ausgegeben. Generell kann man eine bestehende APEX-Anwendung aber jederzeit mit einer beliebigen APEX-Anwendung ersetzen, da immer die komplette Anwendung ersetzt wird.

Vorteile eines gut geplanten Rollouts

Durch gute Vorbereitung wird das Risiko von Fehlern bei einer Installation oder einem Update stark minimiert. Die Installation wird transparenter und es wird viel Zeit gespart. Sowohl die Anwendung als auch das Datenmodell können aus einer Hand ohne weitere Tools über die Weboberfläche installiert werden. Auch durch sinnvolle Kommentare und Prüfungen ist zu jedem Zeitpunkt klar, in welchem Zustand sich eine Anwendung befindet.

Version der Anwendung

Einer der wichtigsten Punkte bei einem Rollout-Prozess und generell in der Softwareentwicklung ist eine gut durchdachte Versionsverwaltung und Definition. Dabei sollte man genau festlegen, wie sich diese zusammensetzt und benutzt wird. Dabei bietet sich folgendes bewährtes Schema an:

Version in der Form: A.B.C.D (Beispiel: 1.3.5.16)

- A (1) => Dies ist das Major Release und ändert sich nur bei einer Neuentwicklung oder einem kompletten Refactoring.
- B (3) => Der zweite Teil der Versionsnummer wird bei neuen Features in der Anwendung erhöht.
- C (5) => Der dritte Teil ändert sich bei jeglicher Änderung im Datenmodell.
- D (16) => Der vierte Teil ändert sich bei jeglicher Änderung in der Anwendung.

Eine solche Versionsnummer sollte sowohl in den Eigenschaften der APEX Anwendung eingetragen werden, als auch im Datenmodell vorhanden sein. Innerhalb des Datenmodells sollte eine Tabelle mit Namen <APEX Anwendungsname>_VERSION vorhanden sein, in welcher die Version eingetragen wird. Diese Version kann nun bei Installation oder Update einer Anwendung benutzt werden um über entsprechende Validierungen sicherzustellen, dass eine Anwendung nur bei entsprechend vorhandener Vorversion installiert werden kann.

Name	
Anwendung:	46242
* Name	Sample Application
Aliasname für die Anwendung	DEMO_APP
* Version	1.2.3.4
Bildpräfix	/i/
Medientyp	
Proxyserver	
* Parsing-Schema	OLIVERLEMM

Abbildung 3: Version und weitere Eigenschaften einer APEX Anwendung

Neben der Version sollten folgende Eigenschaften noch mit in dieser Tabelle hinterlegt werden:

- application_date => Information der letzten Änderung in der APEX Anwendung.
- db_model_version => Hier gehören die ersten 3 Teile der Gesamtversion rein: .B.C (siehe oben)
- application_version => Dies umfasst die komplette oben beschriebene Version
- application_id => Application ID der APEX Anwendung (:APP_ID)
- apex_version => select version_no from apex_release
- nls_db_charset => select value from nls_database_parameters where parameter = 'NLS_CHARACTERSET'
- plsqli_web_toolkit_version => owa_util.get_version()
- servername => owa_util.get_cgi_env('SERVER_NAME')
- http_host => owa_util.get_cgi_env('HTTP_HOST')

- inserted_date => Installationsdatum
- inserted_user => Installationsbenutzer

All diese Eigenschaften werden bei Installation oder Update der Anwendung gespeichert. Zusätzlich wird bei jedem Login in die Anwendung überprüft, ob alle Angaben die bei der Installation oder dem vorherigen Login gesetzt wurden auch noch übereinstimmen. Sobald sich ein Parameter unterscheidet, wird in der Tabelle ein neuer Eintrag erstellt. So lässt sich zum Beispiel ganz genau feststellen, ob sich neben der Installation Eigenschaften an der Umgebung geändert haben und dies zur Fehleranalyse auswerten. Nebenbei kann man auch ganz genau sehen, welche Version an welchem Datum installiert wurde.

Neben der Benutzung der Version bei Installation/Update sollte diese als Label oder Kennzeichen in einer Versionsverwaltung der Dateien mit abgelegt werden. Auch Releasenotes und weitere Dokumente sollten die entsprechende Version beinhalten um den jeweiligen Zusammenhang zu gewährleisten.

Allgemeine Vorgaben

Neben der Version sollte man auch eine klare Struktur innerhalb der Skripte benutzen. Ein Namensmodell in der Datenbank sollte in dieser oder ähnlicher Form vorhanden sein:

- Trigger => <Name der zugehörigen Tabelle oder View>_TRG
- Sequenzen => <Name der zugehörigen Tabelle oder View>_SEQ
- Packages => _PKG
- Tabellen => <Name in Einzahl mit maximal 28 Stellen>
 - Spalten => <Präfix: Kürzel der Tabelle 3-4 Stellen>_
 - Index => <Name der zugehörigen Tabelle oder View>_IDX
 - Foreign Key/Primary Key/Unique Key => <Name der zugehörigen Tabelle oder View>_FK / _PK / _UK
- Funktionen => _FN
- Prozeduren => _PRC

Bei den Dateien sollte man eine solche Erweiterung benutzen:

- <Objektname>.create.sql => Erstellen eines Objekts
- <Objektname>.update.sql => Aktualisierung von Daten
- <Objektname>.delete.sql => Löschen eines Objekts
- <Objektname>.insert.sql => Einfügen von Daten
- <Objektname>.alter.sql => Änderung eines Objekts

Für die Datei der Anwendung

- <Anwendungsname>_<Charset der Exportdatei>.sql => Anwendungsdatei
- <Anwendungsname>_<Charset der Exportdatei>_img.sql => Grafiken
- <Anwendungsname>_<Charset der Exportdatei>_css.sql => CSS Dateien
- <Anwendungsname>_<Charset der Exportdatei>_static.sql => Statische Dateien

Es kann ggf. zur Übersicht beitragen, vor dem Hochladen der Datei ins APEX Repository den Dateinamen um die Versionsnummer zu erweitern, so kann man innerhalb des APEX Repository direkt sehen um welche Version es sich handelt.

Weiter sollte man darauf achten, dass in keinem Skript ein Schemabezeichner vorhanden ist um die Flexibilität zu gewährleisten. Zusätzlich sollte man die Reihenfolge der Skripte und die entsprechenden Abhängigkeiten beachten, damit zum Beispiel eine Änderung am Datenmodell vor einer Datenaktualisierung stattfindet. Komplexe Änderungen an einzelnen Objekten sollten ggf. in einem Migrationsskript pro Objekt gekapselt werden. Bei Objekten wie Views mit vorhandenen Triggern sollte darauf geachtet werden, dass eine Neuerstellung des Views auch den aktuellen Trigger benötigt. In solchen Fällen kann es sinnvoll sein die View und den Trigger immer zusammen zu speichern, um die Fehleranfälligkeit zu minimieren.

Falls das Datenmodell bzw. die Datenskripte nicht in APEX gekapselt werden sollte man Gesamtskripte erstellen, die als einziges die Eigenschaft „spool on“ und „spool off“ besitzen, um eine Logdatei zu schreiben. Eine solche Datei sollte nach jeder Installation/Update mit gespeichert werden und bei Fehlern zur Analyse dienen.

Ablauf eines Rollouts in APEX

Der komplette Ablauf eines Rollouts beinhaltet folgende Schritte:

1. Festlegung der neuen Features
2. Festlegung der neuen Version
3. Festlegung eines Zeitrahmens für Entwicklung, Test und Installation
4. Entwicklung auf einer Entwicklungsumgebung inklusive Versionierung aller Objekte die geändert werden.
5. Test der Entwicklung auf einer Testumgebung
6. Zusammenstellung der Skripte
7. Erstellung der Validierungen bzgl. der Vorversion
8. Sperrung der Testumgebung
9. Kopie der Produktion inklusive Daten auf die Testumgebung (vorher Entfernung aller anderen Objekte auf der Testumgebung)
10. Durchführung eines Update Testlaufs
11. Entsperrung der Testumgebung
12. Kurztest der Testumgebung sowie Überprüfung der Logdateien danach wieder Sperrung
13. Beantragung eines Zeitfensters für die Produktionsinstallation
14. Sperrung der Produktion
15. Sicherung der Daten
16. Sicherung der Datenbankobjekte
17. Sicherung der Anwendung
18. Installation der neuen Anwendung
19. Überprüfung der Logdateien
20. ggf. Vergleich von Testinstallationsobjekten & Produktion durch Schemadiff
21. Freischaltung der Produktion und Testumgebung

Tools

Um innerhalb eines Rollout-Prozesses problemlos arbeiten zu können gilt es sicherzustellen, dass alle geänderten Objekte in einem Versionierungssystem wie CSV, PVCS, oder Subversion abgelegt werden. Diese Systeme sollten alle Skripte, sowie Anwendung und Dokumentation zur jeweiligen Version sichern.

Entsprechende Tools wie TOAD, plsql developer und SQLDeveloper bieten sich zur clientseitigen schnelleren Datenbankentwicklung an und bieten teilweise die direkte Integration in obige Versionierungstools. Neben der Versionierung aller Objekte kann man sich mittels eines Schemadiffs zur Testumgebung auch einen Überblick machen wie groß die Änderungen sind und ob alle Objekte enthalten sind.

Für die Entwicklung eines Datenbankmodells empfehlen sich auch Tools wie Datamodeler oder Rational Rose aus denen komplette Datenmodelle generiert werden können. Hier sollte man sich aber genaue Gedanken machen wie man gewisse Änderungsskripte festhalten oder erzeugen kann.

Kontaktadresse:

Oliver Lemm

MT-AG

Balcke-Dürr-Allee, 9

D-40882 Ratingen

Telefon: +49 (0) 2102-309 61-0
Fax: +49 (0) 2102-309 61-10
E-Mail: oliver.lemm@mt-ag.com
Internet: www.mt-ag.com