Don't Worry Be Happy! Automate Your Environment with Ease!

Noelle Stimely
University of California, Berkeley
USA

Keywords:

Oracle, automation, RMAN, Windows, Linux.

Introduction

How many times do you wish there were more hours in the day? A ticket comes to you for a quick refresh of a dev/test system while you are in the middle of an issue or a large project? This presentation hopes to introduce methods to automate an environment. What to look at and what to do so there will hopefully be more time in your day. Automation is the key to a happy DBA. The goal of this presentation is to show how to automate or at least get down to one line backups/refreshes and other tasks which take time out of the day. We all have our own set of home grown scripts or OEM to alert for problems. I am hoping to offer a new set of scripts or ideas on how to further automate an environment. Some of the automations covered in this presentation are one line database refreshes, backups, monitoring and many more activities critical to the day to day job.

Why Automate?

Those of us who have been DBAs for a significant length of time understand the value of doing more work in less time.   We automate to save time and limit errors than can occur due to human intervention. Manage the environment instead of having it manage you. Know about and understand performance problems before users are impacted. Have the ability to proactively plan for growth or to manage future issues we all know will come up. Get ahead of the curve.

Standardize your environment. This an absolute must. It's at one quarter to one third of the job.

We reduce downtime and we work smarter.

What to Automate?

As database administrators we need to automate as much of our environment as possible to simplify the job. One of the most important areas which needs to be automated is backups. The survival of a DBA depends on the quality of our backups. If we do not have a backup we do not have a job! In designing a backup strategy there are many things to consider. First and foremost is which type of backup is to be performed. Should we be doing the standard weekly full/daily

incremental or more? Hot/cold or a combination? Are table exports needed? We need to work with the application owners to determine the choices and decisions to be made in designing the architecture. After the design has been completed then we can decide on how to automate the process. The environment will dictate the choice of tool (Microsoft Windows vs. Linux/Unix). Also budget will significantly impact this decision.

The tools used can range from scripting with RMAN to purchasing a commercial tool such as Commvault (uses RMAN API). Most all of these tools can be used individually or in combination then scheduled via cron or a scheduler.

- Database Backups
    - Which type are to be performed?
    - Full/incremental/exports/hot/cold
    - Scripting with RMAN
    - Automation via third party tools, cron or a scheduler

After backup decisions have been made thought needs to be put into automating as much of the restore process as possible so in the middle of the night when the unthinkable happens it can be completed without much thought. Also, monthly or at least quarterly test restores of backups should be done to a DR site for process verification.

- Automating Restores
    - Making restoring from a backup an absolute no brainer!
    - What needs to be restored?   Database
                                     Individual tables
                                     Schemas
                                     Other DB structures,
                                     (control files, logs,
                                     tablespaces, etc.)

    - Scripting automated restores to test the validity of a backup
    - Planning for the unthinkable.

Another process, which should be automated, is the refresh of databases or schemas. There are many times application developers and testing teams need to have dev/test databases or schemas refreshed from production for further code testing. How many times have you been working on an intense project only to be interrupted for a database or schema refresh? These refreshes can take precious minutes or hours out of a day and should be automated. When designing an automation strategy certain questions need to be addressed.

- Database Refreshes
    - How often do refreshes need to be performed?
    - Application testing and development
    - Database patching and upgrades
    - Types of refreshes: Database
                              Schema
                              Table

- Performing refreshes   Third party tools
- Script the old fashioned way

When designing an automation strategy for refreshing databases and/or schemas certain items must be taken into consideration. These items are the environment upon which the refresh is taking place, environmental variables, scheduling mechanism and design structure.

I want to provide a moment of pause for those in Microsoft shops. PowerShell is a very powerful tool but anyone who uses it should make sure there is someone on staff who understands it (and Active Directory) inside and out. Ensure that backups are rock solid. If someone makes a mistake, you will need them…and it will happen.

Environmental Considerations   linux/unix   bash/ksh/csh shells
                                            managing global variables
                                            managing profiles
                                             Windows   PowerShell (be careful)
                                             managing .bat files
                                            Windows Scheduler

Major considerations when designing for automation are items such as the accounts which will be used to drive the scripts/processes, security level needed by these accounts, global variables and settings needed to run the scripts, alias needed and how large is the environment. Due to all the recent hacking activities taking place worldwide security is a primary concern. Service accounts used to perform database maintenance activities need to operate on the principle of least privilege and be locked down as much as possible. There are many moving parts in setting up a fully automated environment and all needs to be taken into serious consideration which includes working intensely with the application owners, system administrators and most importantly getting buy in from management. If it is possible to automate mundane tasks a DBAs life can be made much less stressful.

Preparing environment for automation

- Accounts to be used (Oracle, service account, individual)
- Security considerations
- Global environment and account settings.
- Configuration of aliases
- How large is the environment: Number of instances
                                          Number of servers/VMs
                                          Network performance
-The mechanics   Sendmail (it's still around)
                                          Outlook

The hallmark of a fully automated environment include notifications which are great, but care needs to be taken on the number and frequency of such events. Considerations such as who is to be notified, what, when and how needs to be thoroughly deliberated.

Notifications   - Decide what the notification thresholds should be.
                                                    - Too many, too often will drive you nuts.
                                                    - Understand patterns of usage and growth.
                                                    - Establish when or whom to notify and which level.
                                                    - Consult with your system admin.

Consideration of what needs to be monitored needs to be decided as well. As a DBA one of the main aspects of a system needing intense monitoring is the underlying OS disk substructure.

Disk space monitoring   linux/unix   Is ASM being used?
                                                              Script to monitor usage
                                                              Cron jobs, Fix-it tickets
                                   Windows   PowerShell
                                                              Nagios

Oracle "housekeeping activities"

                                   -Frequent checking of the Oracle alert log for errors.
                                   - Other error checking: Remediating errors
                                                                    Process

                                   - Regular log rotation: Audit logs
                                                                    Listener log, etc.

                                   - Keeping the system administrator happy:
                                    Removing core dumps, old install files, old or unneeded
                                    exports or backups; monitor or remove old archives

                                   - Monitoring for further errors or issues:
                                    Which errors? What level for notification?
                                    Automated fixing/ticket creation

As an attendee of this presentation, I am going to share with you a process we use to refresh database schemas/tables in one line. The process involves the configuration of muliple environmental varirables and objects, but once setup the process makes refreshes a simple process. The profiles/scripts/Readme will be included as a download with the paper/presentation.

Scripting Refreshes in one line or less

- Refresh Schema   - Which type? Copy: How does the environment need to be
                              configured to conduct the refresh? Where do the global
                              variables need to be set?
- Copy table(s)

Create the shell script needed for refresh!

- What to refresh: Schema
  Table
  Database

- How to refresh: There are many ways! - shell script (best)
  - tool (not my favorite)

Elements of a shell script: Set global/local profile to environment
  Set variables
  edp.sh – Exports database with Data Pump
  idp.sh – Imports database with Data Pump

Refreshing a Schema - Process

Script will execute the following steps to refresh a schema:
- If the target is a production instance, assert that the schema(s) are empty, otherwise abort.
- Check source schema(s) for any LONG or LONG RAW columns. If present, abort. LONGs cannot be queried over db link.
- Scrub the schemas, excluding tables in exclusion file
- Import the schemas from a remote database source, excluding tables in exclusion file. The import is done using idp.sh with "-s" option.
  Note that the import will always exclude:
  - creating the schema user itself
  - role and system grants to the user
  - private database links (Scrub also excludes database links since it uses all_objects as driver)
  - EXCEPTION: user will be created if doing a single schema import and user does not exist. In that case, the system and role grants to the user will be executed. The user will have the same password as the source.
- Compile invalid objects in target schema(s)
- Call cronsql.sh to process a post-refresh sql script

Script Usage

- Refresh target with copy of schemas from source
- Refresh with remapping of schemas
- Refresh with exclusions
- Create an exclusion file in $HOME/oracle/etc named refresh_<targetdb>_<schema>_exclusions.lst. The file should contain each table to be excluded on a separate line. If you also specify the "-a <app>" option, the exclusions file must be located in $HOME/oracle/<app>/etc.
- Run script with "-x" option

Profiles/Scripts/Readme Attached in zip file

Contact address:

Noelle Stimely
University of California, Berkeley
225-09 2195 Hearst Avenue
Berkeley CA 94720
Phone: 1+ 510 664 4354
Mobile: 1+ 925 915 7390
Email   nstimely@berkeley.edu