



oTN-2009-01

openlab Summer Student Report 2009



Grid Control and Change Management

Syeda Hifsa Kazmi
Supervisor: Christopher Lambert
20th August 2009
Version 1

Distribution:: **Public**



Table of Contents

Acknowledgements	i
1 Introduction	1
1.1 Overview	1
1.2 What is OEM 10g Grid Control?.....	1
1.3 Architecture of OEM 10g Grid Control	1
1.4 What is Change Management System?	<u>3</u>
1.5 Need for this Project.....	4
2 Project Design and Implementation	3
2.1 Getting Data from EM10g Repository	3
2.2 Perl Script GCCM.pl	7
2.3 Feeding Changes to CM	8
3 Test Runs	9
4 Bibliography.....	10



Acknowledgement

I am very grateful to **Juan Manuel Guijarro, Pablo Martinez Pediera, Daniel Lenkes** and all other members of IT-DES group for their help whenever I needed.

Special thanks to my supervisor **Christopher Lambert** for his helpful guidance throughout the project. 😊

Its been pleasure working here!



1 Introduction

1.1 Overview

This report is to explain my work as part of CERN openlab Summer Student Programme 2009. During my stay at CERN, I have been working in IT-DES-DIS section and the title of my project was “Grid Control and Change Management”.

This project was aimed at writing a script which uses Oracle Enterprise Manager (OEM) 10g Repository as data source to extract the change events related to IT-DES group and feed them into Change Management System (CM).

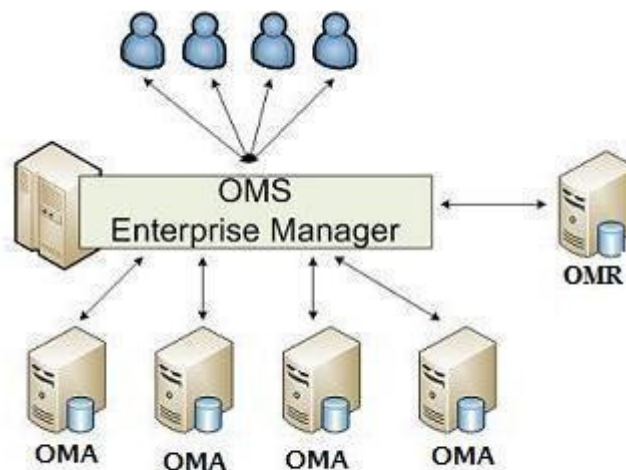
1.2 What is OEM 10g Grid Control?

Oracle Enterprise Manager Grid Control is a tool to monitor and manage multiple instances of Oracle deployment platforms; as well as for management and monitoring of other platforms such as Microsoft .NET, Microsoft SQL Server, NetApp Filers, BEA weblogic and others. Partners and IT organizations can build extensions to Oracle Enterprise Manager, and make them available to other Enterprise Manager users via Oracle Enterprise Manager 10g Grid Control Extensions Exchange.

1.3 Architecture of OEM 10g Grid Control

The architecture of the OEM for Grid Control has three distinct components:

1. Oracle Management Agent or OMA
2. Oracle Management Server or OMS
3. Oracle Management Repository or OMR



Basic Architecture of OEM 10g Grid Control

1.3.1 OMA

The OMA runs on the target host and collects information on the hardware, operating system, and applications that run on the target.

1.3.2 OMS

The OMS runs on one or two servers and collects the data generated by the OMAs. The OMAs push the information to OMS and OMS aggregates the collections into the repository. The OMS also acts



as the user-interface — by generating web-pages for database administrators to view the status of systems and services.

1.3.3 OMR

The OMR comprises an instance of the Oracle database that stores the data collected by the OMS. Installers can make the OMR highly available or fault-tolerant by running it on an Oracle RAC instance across multiple nodes.

1.4 What is Change Management System?

Change Management System or CM in short, is an application written for IT-DES group to log and plan the changes in their systems. It has two parts: a command line interface and a web application.

The command line interface is to let the DES people to add some entries quickly from the Linux machines where they are working on and to give them a tool which can be built into other applications to do automated logging.

The web interface was designed to give the possibility to add changes from other operating systems, to be able to modify an entry, to list and search in requests, to start discussions about a change, to plan future changes and to offer an RSS service.

1.5 Need for this Project

This project is part of implementation of ITIL best practices in IT-DES group at CERN. The aim of the project is to properly log changes detected by OEM 10g Grid into the Change Management System in order to speed up root cause analysis and problem resolution.



2 Project Design and Implementation

- Retrieving the change events from EM10g Grid Control repository views or tables (if required information is not available in views)
- Format the data from obtained from EM10g API view.
- Feed the formatted output to Change Management (CM) tool

2.1 Getting Data from EM10g Repository

The process of getting relevant information from EM10g Repository can be subdivided into following steps.

1. Determine which change event should be recorded in CM database. For this project I worked on following three events.
 - i. Target Down
 - ii. OMA Down
 - iii. Target Blackout
2. Identify EM10g repository views and tables (if required data is not available in views) related to a change event.
3. Form the SQL select statements using the views and tables identified in step 2.

Note: User should be careful when using repository tables in SQL statements because the tables may change in future versions of EM.

2.1.1 SQL statements for Events

2.1.1.1 Target Down

Change Description:

Getting the list of targets which were down in last 24 hours.

Related Views and Tables:

mgmt\$availability_history
mgmt\$group_members

SQL for the Change:

```
SELECT availability_history.target_name,  
       availability_history.target_type,  
       availability_history.availability_status,  
       availability_history.start_timestamp,  
       availability_history.end_timestamp  
FROM mgmt$availability_history availability_history,  
     mgmt$group_members group_members  
WHERE lower(availability_status) LIKE 'target down'  
AND start_timestamp BETWEEN sysdate-1 AND sysdate
```



```
AND
availability_history.target_name=group_members.target_name
/*Limit the results of query to IT-DES Group*/
AND group_name='DES Databases'
```

2.1.1.2 OMA Down

Change Description:

Getting the list of targets on which OMA was down in last 24 hours.

Related Views and Tables:

mgmt\$availability_history
mgmt\$group_members

SQL for the Change:

```
SELECT availability_history.target_name,
        availability_history.target_type,
        availability_history.availability_status,
        availability_history.start_timestamp,
        availability_history.end_timestamp
FROM mgmt$availability_history availability_history,
      mgmt$group_members group_members
WHERE lower(availability_status) LIKE 'agent down'
AND start_timestamp BETWEEN sysdate-1 AND sysdate
AND
availability_history.target_name=group_members.target_name
/*Limit the results of query to IT-DES Group*/
AND group_name='DES Databases'
```

2.1.1.3 Target Blackout

Change Description:

Getting the list of databases which were blacked out in last 24 hours.

Related Views and Tables:

mgmt\$blackout_history
mgmt\$availability_history
mgmt\$blackouts blackouts
mgmt\$group_members
sysman.mgmt_blackouts

SQL for the Change:

```
SELECT availability_history.target_name,
        availability_history.target_type,
        availability_history.availability_status as
        "Availability Status",
        blackout_history.blackout_name,
```



```
blackouts.created_by,
mgmt_blackouts.last_updated_by,
/*Note: Last_updated_by field may not be available
in table sysman.mgmt_blackouts in future versions,
so SQL statement must be modified accordingly.*/
blackouts.status as "Blackout Status",
blackout_history.start_time,
NVL(blackout_history.end_time,'Not Ended Yet') as
END_TIME,
NVL(blackouts.description,'Not Available') as
Description,
blackouts.reason
FROM mgmt$blackout_history blackout_history,
mgmt$availability_history availability_history,
mgmt$blackouts blackouts,
mgmt$group_members group_members,
sysman.mgmt_blackouts mgmt_blackouts
WHERE availability_history.start_timestamp BETWEEN
sysdate-1 AND sysdate
AND blackout_history.start_time BETWEEN sysdate-1 AND
sysdate
/*Note: OEM Restrictions:
1.Duration of a blackout cannot be less than 1 min.
2.The difference between two consecutive blackouts
should be at least 5 minutes greater than the blackout
duration.
*/
AND (ABS((availability_history.start_timestamp -
to_date(blackout_history.start_time))*24*60)-
blackouts.duration) < 5
AND lower(availability_history.availability_status)
='blackout'
AND availability_history.target_name =
blackout_history.target_name
AND blackout_history.blackout_guid =
blackouts.blackout_guid
AND availability_history.target_name =
group_members.target_name
/*Limit the results of query to IT-DES Group*/
AND group_name='DES Databases'
/*Note: Last_updated_by field may not be available in
table sysman.mgmt_blackouts in future versions, so SQL
statement must be modified accordingly.
```




```
*/  
AND mgmt_blackouts.blackout_guid =  
blackout_history.blackout_guid  
ORDER BY availability_history.target_name
```

2.1.2 Saving SQL to GCCM_Data.xml

The SQL statements which would be executed by the GCCM.pl script are saved in an XML file named GCCM_Data.xml.

I have chosen to store SQL in XML file for two reasons:

- To separate data from code
- It would be easy for a user to add SQL statements for new events in future.

The format for saving SQL statements in GCCM_Data.xml file is as follows:

```
<EVENT>  
  <NAME></NAME>  
  <DESCRIPTION></DESCRIPTION>  
  <SQL></SQL>  
</EVENT>
```

e.g. if a user wants to add an event related to targets which have been down in last 24 hours, the entry in GCCM_Data.xml would be like this:

```
<EVENT>  
  <NAME>TARGET DOWN</NAME>  
  <DESCRIPTION>  
    List of targets which were down in last 24 hours  
  </DESCRIPTION>  
  <SQL>  
    <![CDATA[ SELECT availability_history.target_name,  
                      availability_history.target_type,  
                      availability_history.availability_status,  
                      availability_history.start_timestamp,  
                      availability_history.end_timestamp  
    FROM mgmt$availability_history availability_history,  
          mgmt$group_members group_members  
    WHERE lower(availability_status) LIKE 'target down'  
    AND start_timestamp BETWEEN sysdate-1 AND sysdate  
    AND availability_history.target_name =  
        group_members.target_name  
    /*Limit the results of query to IT-DES Group*/  
    AND group_name='DES Databases'
```



```
]]>  
</SQL>  
</EVENT>
```

2.1.3 Possible Enhancements

GCCM_Data.xml file can further be extended to include SQL statements related to:
Configuration changes
Database schema changes - possibly using dictionary baselines EM functionality

2.2 Perl Script GCCM.pl

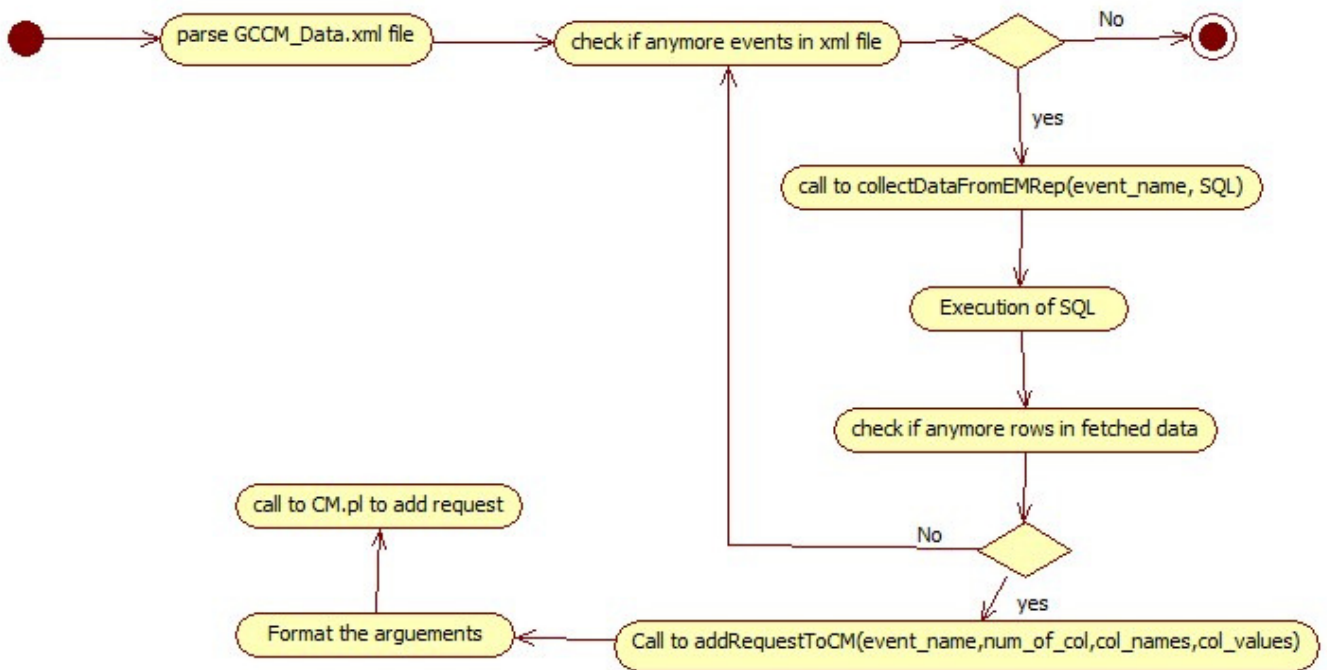
For this project, perl was chosen because this script is going to run UNIX machines and perl is easy to learn and maintain.

This script is written in perl v5.8.5 and it has follow modular dependencies:

- XML::DOM
- DBI

GCCM.pl is the perl script which parses GCCM_data.xml file, picks SQL statements one by one and then execute them to fetch data from EM10g repository. It is assumed that this script would be scheduled to execute once in 24 hours through Cronjobs. Therefore, all the SQL statements in GCCM_Data.xml select the data from last 24 hours.

After fetching the data from EM repository, addRequestToCM() method is called which puts each row from the execution of SQL to CM tool after formatting it.



Activity Diagram for GCCM.pl



2.3 Feeding Changes to CM

When an entry is made in CM database by GCCM.pl, different parameters are set as follows:

- Submitter: The default user name is used if change is not created or updated by a user
- Entity: unknown
- Project: gccm
- Category: Batch
- Prefix of Description: Value of <NAME> tag in GCCM_Data.xml
- Description of change: Column values fetched from select statement

The screenshot shows the CERN IT-DES Change Management interface. The header includes the CERN logo and the text "CERN IT-DES Change Management". Navigation links for "Home", "RSS", "Timeline", "Admin", and "Logout" are visible. The main content area is titled "List of all requests (batches also)" and contains a table of "Change Requests". The table has columns for Id, Submitter, Entity, Project, Scope, Category, Source, Status, Prefix of description, Start date, and Created. Three rows are highlighted in yellow: 2454, 2444, and 2443.

Id	Submitter	Entity	Project	Scope	Category	Source	Status	Prefix of description	Start date	Created
2454	User1	Unknown	GCCM	Unknown	Batch	Syscontrol	done	BLACKOUT TARGET	17. 08. 2009. 12:27:19	17. 08. 2009. 12:27:19
2446	Default	Unknown	GCCM	Unknown	Batch	Syscontrol	done	AGENT DOWN TARG	17. 08. 2009. 12:27:07	17. 08. 2009. 12:27:07
2444	Default	Unknown	GCCM	Unknown	Batch	Syscontrol	done	TARGET DOWN TAR	17. 08. 2009. 12:27:04	17. 08. 2009. 12:27:04
2443	Default	Unknown	GCCM	Unknown	Batch	Syscontrol	done	TARGET DOWN TAR	17. 08. 2009. 12:27:03	17. 08. 2009. 12:27:03
2426	User2	Unknown	ui	Unknown	Batch	SVN	done	SVN commit on I	17. 08. 2009. 11:09:46	17. 08. 2009. 11:09:46



3 Test Runs

During my last week at CERN, I did some test runs of GCCM.pl script. The results of these test runs are summarized in this table

Execution Day	Number of Request Generated for CM
Thu Aug 13 th , 2009	5
Fri Aug 14 th , 2009	4
Mon Aug 17 th , 2009	0
Tues Aug 18 th , 2009	1
Wed Aug 19 th , 2009	0
Thu Aug 20 th , 2009	1
Fri Aug 21 st , 2009	1



4 Bibliography

- Documentation for EM10g Repository Views
http://download.oracle.com/docs/cd/B16240_01/doc/em.102/b40007/views.htm#BACCCEIBI
- OEM10g concept guide.
http://download.oracle.com/docs/cd/B16240_01/doc/em.102/b31949.pdf
- Change Management(CM) tool
https://oracle.web.cern.ch/oracle/Tools/Public/CM/0_2/site/
- CM Documentation
https://oracle.web.cern.ch/oracle/Tools/Public/CM/doc/guide1_2.htm