Essbase ASO Automation Tips

John Ceynar
Hyperion Architect

My Background

- 10+ Years BI/EPM Consulting
- 10+ Years Employee Supporting BI/EPM and/or Software Development
- 10+ Years Hyperion
- Halliburton, Palladium, ThinkFast, Accenture, American Airlines/Sabre Group, Burlington Northern Sante Fe Railroad, Minnesota Mutual Insurance, FedEx, CVS, etc.

Random Thoughts - Disclaimer

- Linux (Exalytics) based tips, but mostly same for windows
- Tips based on my experience, no guarantees
- Interactive - I may throw out some questions to the group
- Very detailed slides and examples

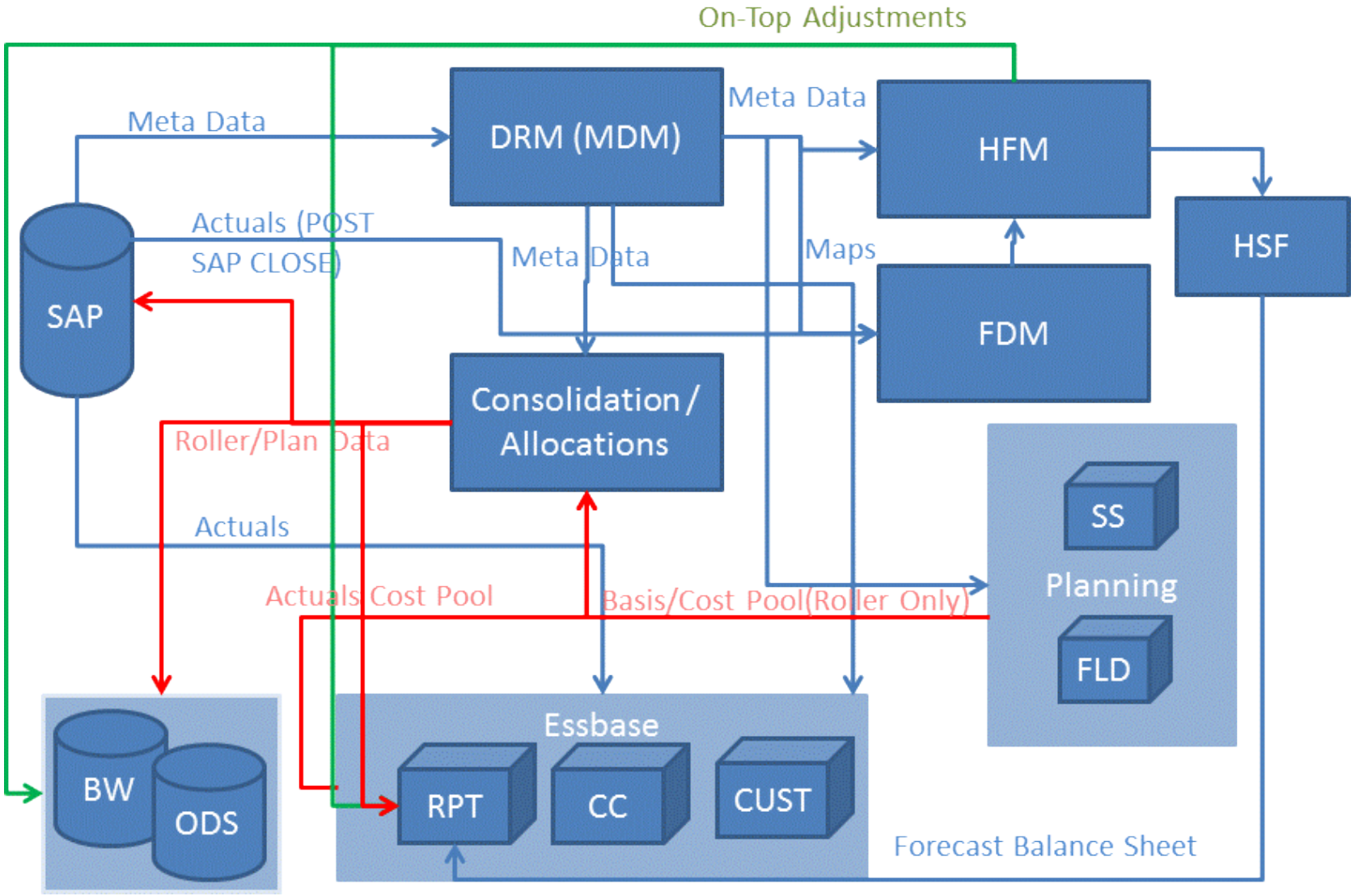
ASO Automation Topics

- Our Environment
- Refreshing Data
  - Speed vs Simplicity vs Up time
- Other Batch Automation Tips
  - Errors, Diff checks, etc.
  - Merging Slices, ReAgging, Backups
- Automatic compares Cross Environments

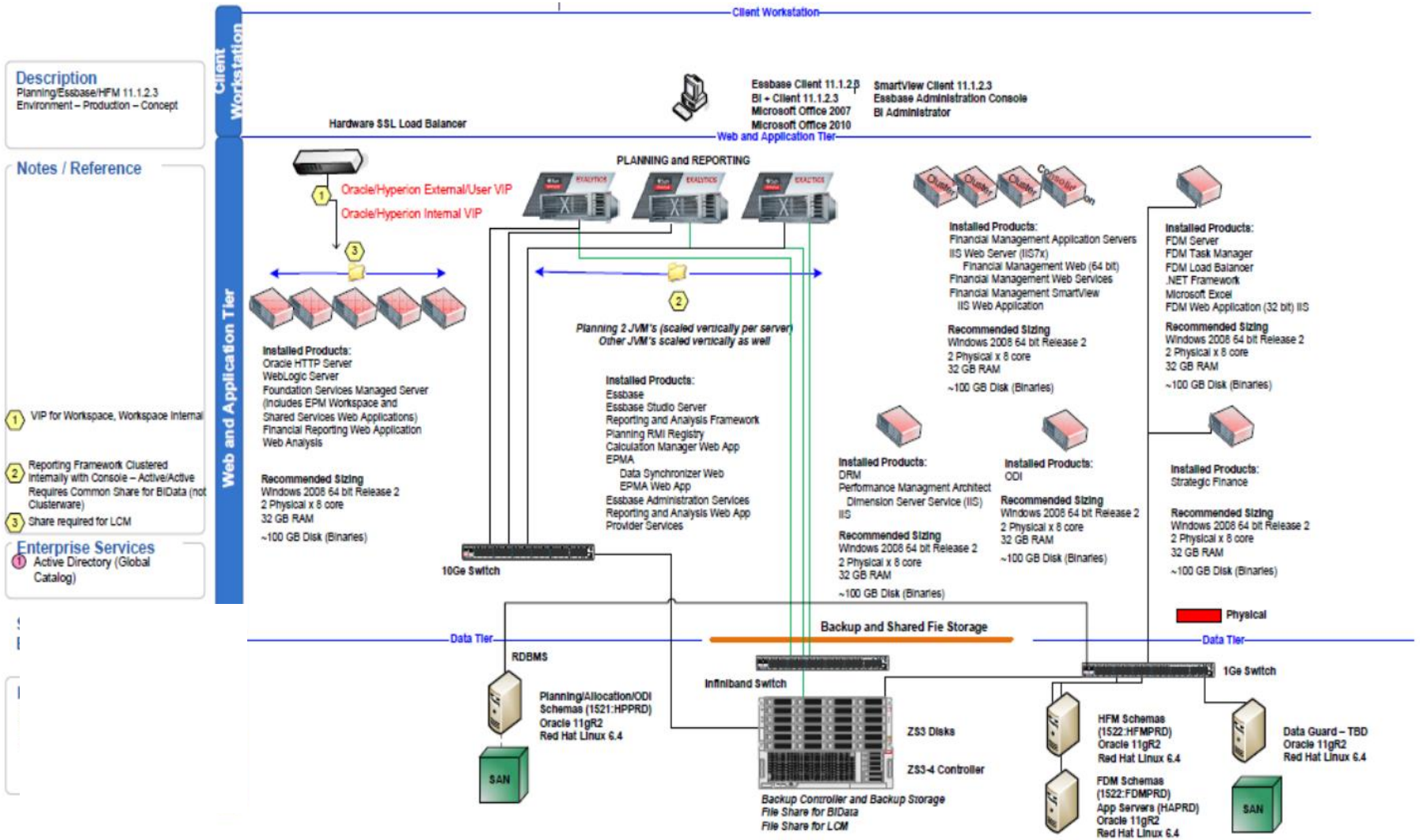
Our Environment

- 11.1.2.3 – Various 500 Patches
- Exalytics and ZFS
- Around 30 Essbase & Planning Apps
- 5000 users, 24x7 global 80 countries
- Dev and Stg are clones of Prod. Slightly fewer servers
- Dev/Stg/Prd – Total 37 servers (reduced by 36) via Exalytics upgrade
- OAM – Single Signon
- SSL everywhere, at http: level
- Made the move to Smart View with the 11.1.2.3 upgrade.

Our Environment Apps



Our Environment - Production



ASO Refreshing Data - Speed vs Simplicity vs Up time

- What is your refresh requirement?
  - Never be zeroed out (like calc script clearing)?
  - Always use slice? Yes, unless small fast load
- Data Load/Refresh Options
  - Slice vs Normal load
  - Clear Report
  - Prior Load File
  - Clear Region MDX

Normal Load vs Slice Load

- “Normal” – has to update all aggregations or drop aggs/rebuild aggs

- Slice loads to mini-cube

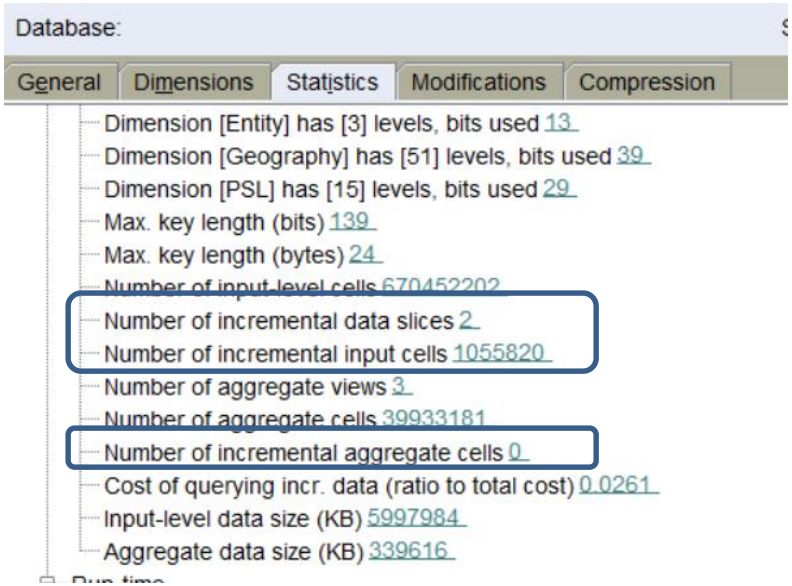
Slice loads – best thing since…

- 11 wonderful characters

```
/**** IMPORT ****/
import database $5.$6 data from load_buffer with buffer_id 1
override values create slice;

echo 'SAP Data Clear & Reload Complete.';
```

- Mini-ASO cubes that users don't know about that automatically merge with main cube to give the correct answer.
- Super Fast Loading



Clear Report

- Report to grab intersections you want cleared
- Never zeroed out (ASO buffer merges clearing file #MI's with load file).
- The report can run in parallel to your source data export as ASO Report scripts can be slower than you'd think.
- MDX report/export would be an option also to create a file.

Clear Report Load MaxL

```
/* Connect To Essbase */
login $key $1 $key $2 on $3;

/* Redirect Output */
spool on to $4/HYP_ESS_MR_D_L_SAP3_$3.log;

/* Only Display Warning & Error Messages */
set message level warning;

/* Start Database */
alter application $5 load database $6;

/**** Clear Aggregates for Faster Loads ****/
/*alter database $5.$6 clear aggregates; */

echo 'Starting SAP Data Load of #MISSINGs to clear the prior load out.';
shell date;

/* Only allow Import to take up 40% of the available resources. */
alter database $5.$6 initialize load_buffer with buffer_id 1 resource_usage 0.4;

/**** CLEAR DATA ****/
import database $5.$6 data
from data_file "'$10/Export_SAPmr_Slice_For_Clear_$3.txt'" using server rules_file 'ClrSlice.rul'
to load_buffer with buffer_id 1
on error write to "'$9/HYP_ESS_MR_D_L_SAP3_ClrMR_$3.err'";

echo 'Starting SAP Data Load.';
shell date;

/**** LOAD DATA ****/
import database $5.$6 data
from data_file "'$7/$8'" using server rules_file 'L_SAP.rul'
to load_buffer with buffer_id 1
on error write to "'$9/HYP_ESS_MR_D_L_SAP3_$3.err'";

/**** IMPORT ****/
import database $5.$6 data from load_buffer with buffer_id 1
override values create slice;

echo 'SAP Data Clear & Reload Complete.';
shell date;

/* Close Spool File & Exit */
spool off;
exit;
```

Usually don't clear your aggregates, usually no need if doing slice load.

Load rule that ignores data column and loads #MI.

Since loading the "Clear file" and the new data in same buffer, the only cleared cells are the truely zeroed out cells.

Prior Load File

- Just use prior load file (have to make sure same month) for knowing intersections to clear.
- Never zeroed out (ASO buffer merges clearing file #MI's with load file).
- Must have more complex logic making sure prior file actually did load. Otherwise, you can miss some cleared intersections.
- Should have Clear Region, etc. script as backup approach for when Prior File approach fails.

Prior Load File - example

```
CurrYrMth=`head -n1 $HYP_ESSInputFilePath/cc_Actuals_Curr_Mth.txt | cut -f2-3 -d, `
echo $CurrYrMth
PriorYrMth=`head -n1 $HYP_ESSInputFilePath/cc_Actuals_Curr_Mth_Prior_SuccLoad_File_$HYP_ESSServer_Mgt_CC_Prod2.txt | cut -f2-3 -d, `
echo $PriorYrMth

if [ $CurrYrMth = $PriorYrMth ]
  then
    echo "Reloading/refreshing Same Month's Data.  Will clear using prior successfully loaded files."
    ######################### CLEAR & RELOAD - BEGIN ##############################
    $HYP_ESSMaxLExe -D $HYP_ESSShellScripts/HYP_ESS_Mgt_CC_D_L_SAP6.mshs $HYP_ESS_PrivKey $HYP_ESSUserID $HYP_ESSPassword $HYP_ESSServer_Mgt_CC_Prod2 $HYP_ES

    #################  Check if any UNEXPECTED ERRORS on ESSBASE LOADS.  ##############################
    grep "Unexpected Essbase error" $HYP_ESSMaxLLogs/HYP_ESS_Mgt_CC_D_L_SAP6_$HYP_ESSServer_Mgt_CC_Prod2.log > $HYP_ESSMaxLLogs/HYP_ESS_Mgt_CC_D_L_SAP5_ERRFO
    grep "essmsh error" $HYP_ESSMaxLLogs/HYP_ESS_Mgt_CC_D_L_SAP6_$HYP_ESSServer_Mgt_CC_Prod2.log >> $HYP_ESSMaxLLogs/HYP_ESS_Mgt_CC_D_L_SAP5_ERRFOUND_$HYP_ES
    grep "ERROR" $HYP_ESSMaxLLogs/HYP_ESS_Mgt_CC_D_L_SAP6_$HYP_ESSServer_Mgt_CC_Prod2.log >> $HYP_ESSMaxLLogs/HYP_ESS_Mgt_CC_D_L_SAP5_ERRFOUND_$HYP_ESSServer
    # -s tests that file exists and is not empty
    if [ -s $HYP_ESSMaxLLogs/HYP_ESS_Mgt_CC_D_L_SAP5_ERRFOUND_$HYP_ESSServer_Mgt_CC_Prod2.txt ]
      then
        echo '*************************************************************************************************
        echo '****** Unexpected Essbase Error on the 029 load.  *********'
        echo '*************************************************************************************************'
        echo $HYP_ESSServer_Mgt_CC_Prod2": Unexpected Essbase Error on the curr mth Actuals data load to Mgt_CC.  Mgt_CC should not be cleared, but might not
      else
        echo $HYP_ESSServer_Mgt_CC_Prod2": No Unexpected Essbase error loading.  Continue."
        cp -p $HYP_ESSInputFilePath/cc_Actuals_Curr_Mth.txt $HYP_ESSInputFilePath/cc_Actuals_Curr_Mth_Prior_SuccLoad_File_$HYP_ESSServer_Mgt_CC_Prod2.txt
        sqlplus $HYP_ESSSQLUserID_HYPBW/$HYP_ESSSQLPassword_HYPBW@$HYP_ESSSQLSID_HYPBW @/d1/Hyperion/BatchJobs/Scripts/HYP_ESS_DAILY_JOB_LOG.sql "$HYP_ESS_Mg
    fi
```

Must be same month file

These 3 checks cover every error I've ever seen

Must be sure no errors loading, else don't update prior file

Clear Region MDX – Best thing since slices…

- Fast, Simple (relatively)
- But, like BSO Calc script clearing, data is gone

```
/* Connect To Essbase */
login $key $1 $key $2 on $3;

/* Redirect Output */
spool on to $4/HYP_ESS_MR_D_L_WD_ClrRegion_All_$3.log;

/* Only Display Warning & Error Messages */
set message level warning;


/* Clears any historical WellDynamics entries in any Scenario - SAP/HFM/Roller. */
/*Logical - creates new slice with offsets to data in the Region. */
alter database $5.$6 clear data in region '{Crossjoin({[Mgmt. Reporting w/o Well Dynamics]},{[1WELDYN_SU]})}';


/******* Notes and Other Options we could have used for individual clearing. *********/
/* - Can use sub vars */
/* - Node specified in Clear Region must be stored.  It can be top level node, then all level zero stored descendants will get cleared.*/
/* - If don't specify dimension, it will clear any stored members in that dimension that has data for the intersection. */
/*alter database $5.$6 clear data in region '{Crossjoin({[Mgmt. Reporting w/o Well Dynamics]},{Crossjoin({[1WELDYN_SU]},{[&LoadRoller]})})}';*/
/*alter database $5.$6 clear data in region '{Crossjoin({[Mgmt. Reporting w/o Well Dynamics]},{Crossjoin({[1WELDYN_SU]},{[ActualUS$-HFM Adj]})})}';*/
/*Physical - merges slices into main cube and clears region in Main Cube.  Slow, 15 min or more.*/
/*alter database $5.$6 clear data in region '{Crossjoin({[Mgmt. Reporting w/o Well Dynamics]},{Crossjoin({[1WELDYN_SU]},{[ActualUS$-HFM Adj]}))}' Physical;'


echo 'Clearing WellDynamics profit centers out of w/o Well Dynamics viewpoint is complete.';
shell date;

/* Close Spool File & Exit */
spool off;
exit;
```

Hardest part is getting sytax right on multiple dimensions

Logical (default) or Physical

ASO Refreshing Data - Summary

- What is your refresh requirement?
  - Never be zeroed out (like calc script clearing)?
    - Then Clear Report or Prior Load File option
    - Else Clear Region MDX
  - Always use slice? Yes, unless small fast load

Other Batch Automation Tips

- Error Checking
  - Empty Source File
  - Load failures (example on Prior Load pg)
  - Error Files, kickouts
- Diff Checks to Prior load file

```
# -s tests that file exists and is not empty
if [ -s $HYP_ESSInputFilePath/cc_Actuals_Curr_Mth.txt ]
 then
  echo
 else
  echo '****Exit - no data to load*****'
  echo $HYP_ESSServer": No SAP source file to load to Mgt_CC." | mail -s "HYP_ESS_Mgt_CC_D_L_SAP.Sh - No SAP source file." $HYP_ESSAdminEmail_2 $HYP_FinUserE
  exit
fi

diff -biw $HYP_ESSInputFilePath/cc_Actuals_Curr_Mth.txt $HYP_ESSInputFilePath/cc_Actuals_Curr_Mth_Prior_SuccLoad_File_$HYP_ESSServer_Mgt_CC_Prod2.txt > $HYP_
if [ -s $HYP_ESSInputFilePath/cc_Actuals_Curr_Mth_Diffs.txt ]
 then
   echo 'Continue, there are new entries in the latest Current Month SAP data file (cc_Actuals_Curr_Mth_Diffs.txt).'
 else
   echo $HYP_ESSServer": No SAP file changes, so skipping Mgt_CC reload." | mail -s "HYP_ESS_Mgt_CC_D_L_SAP.Sh - No SAP changes." $HYP_ESSAdminEmail_2 $HYP_F
   exit
fi
```

Other Tips

- Central parameter file

```
#!/bin/bash
################################################################################################################################
############# ESSBASE APPS and JOB VARIABLES ###############

/u02/Oracle/Middleware/user_projects/epmsystem1/EssbaseServer/essbaseserver1/bin/setEssbaseEnv.sh
/u02/Oracle/Middleware/user_projects/epmsystem1/EssbaseServer/essbaseserver1/bin/startMaxl.sh
HYP_ESSMaxLExe=/u02/Oracle/Middleware/user_projects/epmsystem1/EssbaseServer/essbaseserver1/bin/startMaxl.sh
HYP_ESSShellScripts=/u02/Hyperion/BatchJobs/Scripts
HYP_ESSShellLogs=/u02/Hyperion/BatchJobs/shLogs
HYP_ESSErrPath=/u02/Hyperion/BatchJobs/ErrorFiles
HYP_ESSErrorFilePath=/u02/Hyperion/BatchJobs/ErrorFiles
HYP_ESSMaxLLogs=/u02/Hyperion/BatchJobs/mshLogs
HYP_ESSInputFilePath=/u02/Hyperion/BatchJobs/InputData
HYP_ESSOutputFilePath=/u02/Hyperion/BatchJobs/OutputData

############# PLANNING APPS VARIABLES ###############
HYP_ESSServer_LMK_REVP=Server1
HYP_ESS_LMK_REVP_APP=LMK_REVP
HYP_ESS_LMK_REVP_DB=LMK_REVP
```

- Merging Slices

```
/* Merges all slices into Main Cube.  Also, removes impacted cells that would be zero from main cube. */
alter database $5.$6 merge all data remove_zero_cells;
```

Backups (Merge, Reagg, export)

```
echo '******** Mgt_CC Merge Slices *************'
$HYP_ESSMaxLExe -D $HYP_ESSShellScripts/HYP_ESS_Mgt_CC_D_BACKUP1_MergeSlices.mshs $HYP_ESS_PrivKey $HYP_ESSUserID $HYP_ESSPassword $HYP_ESSServer $HYP_ESSMaxLLogs $HYP_ESS_M

echo '******** Start Mgt_CC ReAgg *************'
$HYP_ESSMaxLExe -D $HYP_ESSShellScripts/HYP_ESS_Mgt_CC_D_BACKUP2_ReAgg.mshs $HYP_ESS_PrivKey $HYP_ESSUserID $HYP_ESSPassword $HYP_ESSServer $HYP_ESSMaxLLogs $HYP_ESS_Mgt_CC_

echo '******** Start Mgt_CC Backup Export *************'
$HYP_ESSMaxLExe -D $HYP_ESSShellScripts/HYP_ESS_Mgt_CC_D_BACKUP1.mshs $HYP_ESS_PrivKey $HYP_ESSUserID $HYP_ESSPassword $HYP_ESSServer $HYP_ESSMaxLLogs $HYP_ESS_Mgt_CC_APP $H

# -s tests that export file exists and is not empty
if [ -s $HYP_ESSOutputFilePath/HYP_ESS_Mgt_CC_D_BACKUP.txt ]
then
  typeset -i CountofCCBackup=0
  curday=`date +%d`
  echo 'CurrentDay: ' $curday
 # Set date variables.
  yyyy=`date +%Y`
  mm=`date +%m`
  dd=`date +%d`

  #### Default file copy and zip
  gzip -f $HYP_ESSOutputFilePath/HYP_ESS_Mgt_CC_D_BACKUP*
  cp $HYP_ESSOutputFilePath/HYP_ESS_Mgt_CC_D_BACKUP.txt.gz $HYP_ESSBackupFilePath/HYP_ESS_Mgt_CC_D_BACKUP_$yyyy$mm$dd.txt.gz
  if [ $curday = '09' ]  #Take monthly copy after close
    then
      cp $HYP_ESSOutputFilePath/HYP_ESS_Mgt_CC_D_BACKUP.txt.gz $HYP_ESSBackupFilePath/HYP_ESS_Mgt_CC_Mth_BACKUP_$yyyy$mm.txt.gz
  fi

  #### Getting the count of backup files
  CountofCCBackup=$(ls $HYP_ESSOutputFilePath/HYP_ESS_Mgt_CC_D_BACKUP* | wc -l)
  echo CountofCCBackup
  if [ $CountofCCBackup > "1" ]
   then
   ## do while / for statement for all additional files except default
   while [ $CountofCCBackup -gt 1 ]
    do

    CountofCCBackup=CountofCCBackup-1
    cp $HYP_ESSOutputFilePath/HYP_ESS_Mgt_CC_D_BACKUP_$CountofCCBackup.txt.gz "$HYP_ESSBackupFilePath/HYP_ESS_Mgt_CC_D_BACKUP_$CountofCCBackup""_$yyyy$mm$dd.txt.gz"

    if [ $curday = '09' ]
      then
        cp $HYP_ESSOutputFilePath/HYP_ESS_Mgt_CC_D_BACKUP_$CountofCCBackup.txt.gz "$HYP_ESSBackupFilePath/HYP_ESS_Mgt_CC_Mth_BACKUP_$CountofCCBackup""_$yyyy$mm.txt.gz"
    fi
    echo CountofCCBackup
   done
  fi

  echo "Start - Removing daily backup files older than 30 days from ZFS."
  # Remove any daily backup files in the shared path older than 30 days.  Leave the monthly f
  find $HYP_ESSBackupFilePath \( -name 'HYP_ESS_Mgt_CC_D_BACKUP_*' \) \-mtime +30 -exec rm {}
```

After 2GB backup export splits off another file. We do a loop so we don't miss when another file is added.

We keep daily copy and month end copy.

Delete daily backups older than 30 days

Automatic Cube Compares Cross Environments

- New MaxL command in 11.1.2.3
- export outline app.db all dimensions to xml_file “’path/FileOutput.xml'";
  - Default is all properties
  - Tree option for only Parent/Child only

Automatic Compares Cross Environments

Function called for every cube

```
################################################################################################################
################################### FUNCTIONS ##################################################################
################################################################################################################

CompareOutlines()
{
$HYP_ESSMaxLExe -D $HYP_ESSShellScripts/HYP_ESS_Util_Outline.mshs $HYP_ESS_PrivKey $HYP_ESSUserID $HYP_ESSPassword $1 $HYP_ESSMaxLLogs $2 $3 $HYP_ESSInputFilePath $
$HYP_ESSMaxLExe -D $HYP_ESSShellScripts/HYP_ESS_Util_Outline.mshs $HYP_ESS_PrivKey $HYP_ESSUserID $HYP_ESSPassword $4 $HYP_ESSMaxLLogs $5 $6 $HYP_ESSInputFilePath $

diff $HYP_ESSOutputFilePath/HYP_ESS_Util_Outline_$2$1.xml $HYP_ESSOutputFilePath/HYP_ESS_Util_Outline_$5$4.xml > $HYP_E      putFilePath/HYP_ESS_Util_Outline_$2$1.di

if [ -s $HYP_ESSOutputFilePath/HYP_ESS_Util_Outline_$2$1.diffs ]
  then
    echo $2$1' Outline xml export files do not match to '$5$4'  files    See $HYP_ESSOutputFilePath/HYP_ESS_Util
fi
}

################################################################################
################################### Main      ##############
################################################################################
rm -f $HYP_ESSOutputFilePath/HYP_ESS_Util_Outline_All.diffs

EssServer1=Server1
EssApp1=FLD_ASA
EssDB1=FLD_ASA
EssServer2=Server2
EssApp2=FLD_ASA
EssDB2=FLD_ASA
CompareOutlines $EssServer1 $EssApp1 $EssDB1 $EssServer2 $EssAp

EssServer1=Server1
EssApp1=FLD_EER
EssDB1=FLD_EER
EssServer2=Server3
EssApp2=FLD_EER
EssDB2=FLD_EER
CompareOutlines $EssServer1 $EssApp1 $EssDB1 $EssServer2 $EssAp
```

Call it for each server comparing

```
login $key $1 $key $2 on $3;

/* Redirect Output */
spool on to $4/HYP_ESS_Util_Outline_$5_$3.log;

/* Only Display Warning & Error Messages */
set message level warning;

/* Start Database */
alter application $5 load database $6;

echo 'Starting export of outline.';
shell date;

/* By default, dumps everything in outline, members, formulas, UDAs, all aliases, etc. - Best for Fu
export outline $5.$6 all dimensions to xml_file "'$9/HYP_ESS_Outline_$5$3.xml'";

/* tree - just dumps outline members, no formul      liases, etc.  - Nice for simple high level compa
/*export outline $5.$6 all dimensions tree to xml      '$9/HYP_ESS_Util_Outline_$5$3.xml'";*/

echo 'Export Complete.';
shell date;

/* Close Spool File & Exit */
spool off;
exit;
```

The magic MaxL