Hyperion Focus 2015

CME – Hyperion Story

Filip Jansky

CME

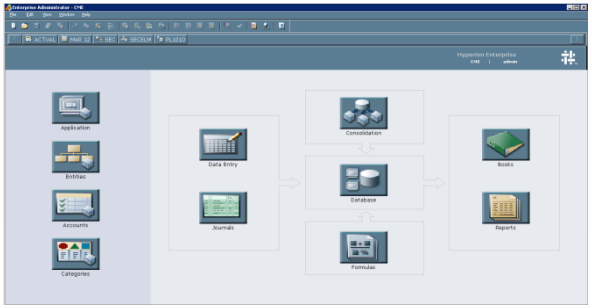
Introduction

- CME – Central European Media Enterprises
  - Media and entertainment company operating leading businesses in primarily six Central and Eastern European markets: Bulgaria, Croatia, the Czech Republic, Romania, the Slovak Republic and Slovenia.
  - Listed in the United States on the NASDAQ Global Select Market and in the Czech Republic on the Prague Stock Exchange

- Filip Jansky - Manager of Reporting Systems & BI at CME
  - Responsible for HFM, Data Warehouse and reporting tools
  - 5 years experience with HFM (previously working with Hyperion Planning)

Legacy

- Hyperion Enterprise
  - approx. 10000 accounts, 8 entity structures (org by period),
  - lots of adjustment entities (also heavily used)
  - data sources:
    - FDM (no ERP integration)
    - Text files
    - Excel - Hyperion Retrieve



  - With increasing amount of entity structures and accounts HyE became quite complicated for maintenance
- 2009/10 – First attempt to implement HFM
  - using HyE chart of accounts and entity structure
- 2010/11 – HFM and DWH (MS SQL Server Based) implementation
  - Defining new CoA using dimensionality
  - Aug 2011 – HFM Implementation partner changed to Forensys
  - Apr 2012 – HFM used as only consolidation system

Original HFM System design

- All 4 custom dimensions used:
  - Segment – 2 structures: Functional & Regio-Functional (one entity could contain multiple functional segments)
  - Intercompany type (3rd party, related party, IC, IC Margin,..)
  - Account attribute/movement
  - Nature
- Only one Entity structure – legal view
- Periodic data loads (movements) + closing balance splits for current noncurrent
- Forecasting not based on legal entities but rather on segments – complex relation between actual and Fct data
- Scenarios for different rates (Act, Bgt, next year's Act,…)
- Automation through HFM task flow

Experiences with Original design

- Biggest challenges
  - Migration from HE (different approach to adjustments, minorities consolidated – previously loaded)
  - large sub-cubes on higher entity nodes (over 150000 units)- slow consolidation times
  - forecast logic
  - Task automation – highest recurrence – daily, time consuming maintenance
- Greatest benefit compared to HyE
  - Less time spend with system maintenance (no year roll-forwards)
  - You don't have to kick users from system during consolidation
  - Transparency
    - where does this consolidated figure comes from? (Nature dimension)
    - Account hierarchies instead of subtotal accounts (general HFM attribute)
    - No adjustment entities
  - Clarity - what is the number I want to report (only one entity hierarchy)

HFM 2014 Redesign

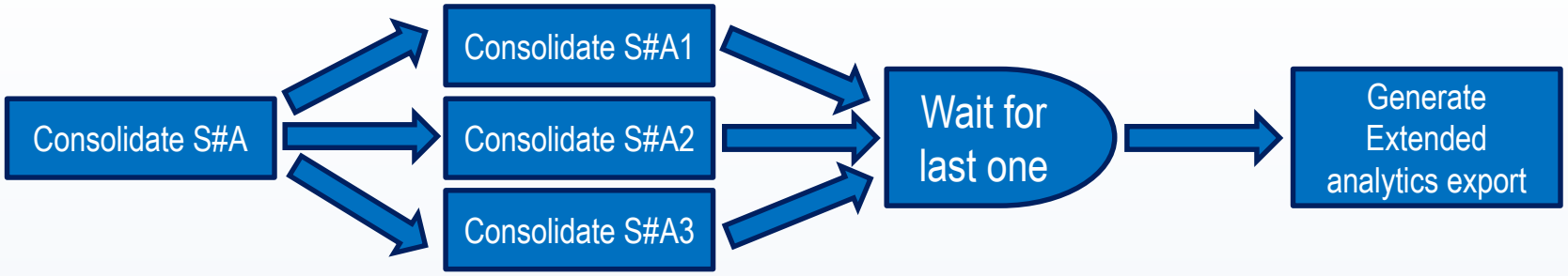
- Flattening Entity structure (removing nodes not required for reporting to reduce consolidation times)
- Changes in Balance sheet account/movement structure (no more loading of closing balance split)
- Segments simplified 1 Entity = 1 Segment
- Forecast structure aligned with actual structure
- Added discontinued business logic (additional scenarios + logic for transformation of source scenarios' data)
- Max Sub cube size reduced to approx. 40000 units

Experiences with redesigned application

- Biggest challenges
  - number of derived scenarios (rate logic + discontinued business) - increase of overall processing time, complexity of scenario impact logic
  - Still using HFM Task Automation
- Benefits compared to previous version
  - Simplified dependency between forecast and actual scenarios (same logic)
  - Simplification of segmental eliminations (1 entity one segment - no need to book IC partner segment information for correct consolidation results)
  - No need of validations between movements and closing balance loads
  - General reduction of data amount - consolidation time down by **85%**

EPM Maestro

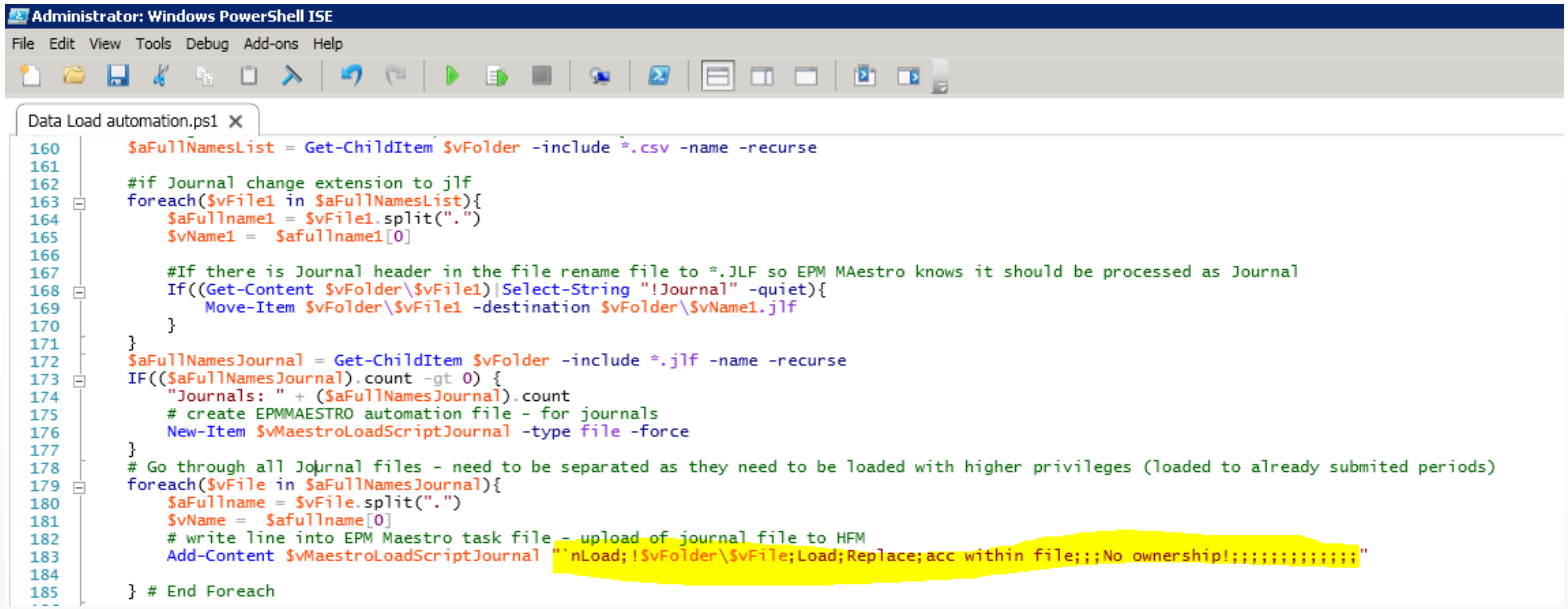
- Implemented September 2014
- First used for automating standard administration tasks (immediately saved some 4 hours/month), scheduled consolidations/exports to DWH
- Immediate relief for HFM administrators -much easier to reprocess multiple scenario/year consolidations/translations
- Redesign of scenario logic to benefit from using of threads in Maestro
  - great benefit when dealing with numbers of derived scenarios
  - you can consolidate/translate these simultaneously reducing the overall processing time



PowerShell

OK - so we automated part of the admin routines how about automating some user tasks?

- Adding PowerShell into play
  - PowerShell is a task automation and configuration management framework from MS, consisting of a command-line shell and associated scripting language built on the .NET Framework.



EPM Maestro + PowerShell

- Ad-hoc run of certain Maestro task flows through allocation routine
  - users had rights to start previously prepared HFM task flows – we wanted them to still be able to run the tasks (yet we wanted to have these task in Maestro and not maintain it on both sides)
  - original task flows replaced with allocate function, allocation routine then contains just a command to start specific PowerShell script

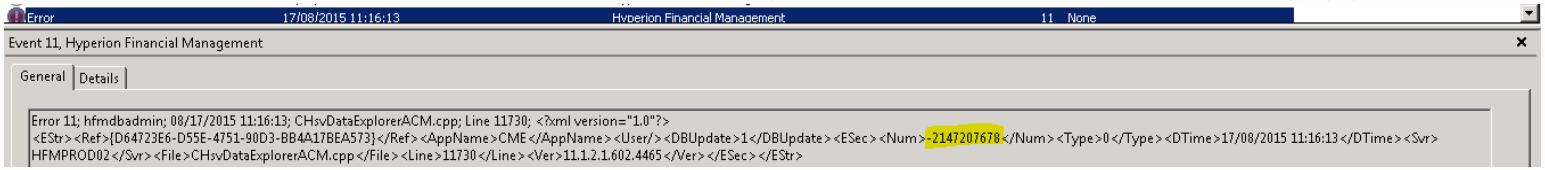
```
SUB Allocate()
  'Allocation routine is used for automation purposes'
  writetofile "Allocate"
  Call SetMembers(1)
  Select Case Entity 'solution needs to be able to support more entities
    'In Case of CZ110CME/Register/2009/Jan - this starts a file with EPM maestro taskflow
    Case "CZ110ENT"
      IF Scenario = "Register"  And HS.Year.Member = "2009" Then
        IF Period = "Jan" then
          vString = "powershell -file "& Chr(34) &"\\HFMPROD01\EPMMaestroTasks\Actual_every_hour.ps1"
          Set WshShell = CreateObject("WScript.Shell")
          WshShell.run(vString)
        ElseIf Period = "Feb" then
          vString = "powershell -file "& Chr(34) &"\\HFMPROD01\EPMMaestroTasks\forecast.ps1"
          Set WshShell = CreateObject("WScript.Shell")
          WshShell.run(vString)
        ElseIf Period = "Mar" then
          vString = "powershell -file "& Chr(34) &"\\HFMPROD01\EPMMaestroTasks\budget.ps1"
          Set WshShell = CreateObject("WScript.Shell")
          WshShell.run(vString)
        End If
      End If
      Exit Sub
```

EPM Maestro + PowerShell 2

- Automation of whole data processing
  - Source systems:
    - Local Data Marts (MSSQL Server) - actual data
    - Journal workflow (custom build system for creating/approving late adjustments) - journals once entities are in submitted state
    - Anaplan (forecasting/budgeting tool) - forecast/budgets
  - Regularly (every minute) checking import folder on HFM server if file(s) present
  - Loading data/journals to HFM
  - Based on file content notify responsible person
  - launch additional processing:
    - Calculation of impacted entities
    - Launch Maestro task flow consolidating/translating/exporting data (in case no error in data load)
    - Prevent from launching same Maestro task if it is already running
  - Archive the loaded file and log

PowerShell - HFM Error notification

- Further use of PowerShell scripts - HFM error notification
  - using event triggered task in windows scheduler not an option -To general - HFM error reference in the body of logged event - can not be the trigger



  - Building email notification in PowerShell
    - Uses a predefined list of HFM error with severity parameter (critical/normal/low) and notification flag(notify/ignore)
    - Running on one server but parsing event logs from whole HFM environment

```
#query - XML statement filtering HFM error entries in last X miliseconds
$query = @"
<QueryList>
  <Query Id="0" Path="Application">
    <Select Path="Application">*[System[Provider[@Name='Hyperion Financial Management']
    and (Level=2) and (EventID=11)
    and TimeCreated[timediff(@SystemTime) &lt;= "$timeaddMS"]]]
  </Select>
  </Query>
</QueryList>
"@
```

```
Foreach ($Server in $Serverlist){$Server;
    $time = get-date
    #Xml filter is much quicker but is not able to analyse message part in detail - so it is used jujst to check whether there are any Error entries in log
    $ErrorX = Get-WinEvent -FilterXml $query -ComputerName $server
    ($ErrorX).Count
    #Check if number of error lines > 0
    IF (($ErrorX).Count -ne 0) {
        #get all error lines from application log related to HFM in last X minutes
    $Errordata  = (get-eventlog -after $time.AddMinutes($timeadd) -logname application -Source 'Hyperion Financial Management' -EntryType Error -ComputerName $server)
        #go through infividual error lines
        foreach($string in $Errordata){
```

Questions?

Thank you!


EPM Maestro


Merlin xl


AMOSCA
Balancing Business and Systems


VertexFusion


keyteach


REPORT
AUTHORITY


INFRATECTS


nymad


ORACLE


epm
FastTrack


Accelatis
Better Performance. Less Guesswork.


UK OUG
UK ORACLE USER GROUP


CXO COCKPIT
WEB-BASED & MOBILE CORPORATE EPM REPORTING


TPH