HFM Administration Tips & Tricks

Kris Beatty

eCapital Advisors
A MANAGEMENT CONSULTING FIRM

eCapital Overview

- **Founded in 2001**
- **Enterprise performance management consulting firm**
- **Headquartered in Minneapolis, MN**
- **Regional offices in St. Louis and Milwaukee**
- **Over 150 performance management customers**
- **eCapital Advisors employees:**
  - Proven record and high customer satisfaction
  - Highly experienced project team
  - Deep Finance and Accounting backgrounds
  - Integrated EPM knowledge

Agenda

- Application Design Considerations
- General Admin Considerations
- Decode Admin Error Messages
- Rules to Perform Common Maintenance
- Cash Flow Design Options
- Q&A

Application Design Considerations

Application Profile

1. Year
2. Period
3. View

System

4. Value dimension, includes currencies

Admin Defined

5. Entity
6. Account
7. ICP
8. Scenario

9-12. Custom 1-4

Application Design Considerations

Year

- Consider number of years you need in application, typically see 15
- Cannot be changed after the application is built

Period

- Avoid weekly or yearly profiles unless required
- If weekly is required consider building separate application for this data

View

- Only Periodic and YTD are stored, QTD is derived on-the-fly

Application Design Considerations

Value Dimension

- Different types of values stored in the application
- Members come with the application and cannot be changed
- <Entity Currency> current entity's default currency, typically contains what is loaded from the local ledger
- <Entity Currency Adj> also in current entity's default currency, and is only used for journal entries
- <Parent Currency> is the currency of the entity's parent
- Can be used to translate into any other currency in your application (use with caution)

Application Design Considerations

Entity

- Legal or management structure of the company
- Avoid Consolidate All on each hierarchy

ICP

- Recommend setting base entities as ICP
- Set flag to ‘R’ to restrict entities creating ICP with self

Scenario

- Defines the data you are collecting

Customs

- Use for items that apply across multiple accounts

General Admin Considerations

- Make application as dynamic as possible
  - Build dynamic account lists by using UD fields
  - Use metadata wherever possible to build hierarchies rather than writing a rule
  - Your customs can be used for more than one purpose
- Leverage business rules to set up functions for common calculations that you can call multiple times in your file
- Be careful not to write a rule that will fill your database with zero values
- Avoid formulas within reporting
- Use one of your custom dimensions to store data type

Sample Custom 4 Data Type

N°		Member	Description English	Is Calculated	SwitchSign ExpFlow	SwitchType ExpFlow	User Defined1	User Defined2	User Defined3	Security Class	SubmissionGroup	Node
0		CUSTOM4										
		DataTypes	Data types	N	N	N						0
		AfterRestat	Total After Restatements	N	N	N						1
		Restated	Restatements journals	N	N	N						1
		BeforeRestat	Total Before Restatements	N	N	N						1
		YearEnd	Year-end journals	N	N	N						1
		BefYearEnd	Total Before Year-end adjustments	N	N	N						1
		TotalElims	Total Eliminations (calc)	N	N	N						1
		Elimin	Eliminations (calc)	Y	N	N						1
		NetInvestCSElims	Net Elim of Invest and Com Stock (calc)	N	N	N						1
		InvestElim	Elim of Invest against Min Inter (calc)	Y	N	N						1
		CSElim	Elim of Com Stock against Min Int (calc)	Y	N	N						1
		CYEarnings	Elim MI of CY Earn against Min Int(calc)	Y	N	N						1
		DistrDiv	Elim MI of Div against Min Inter (calc)	Y	N	N						1
		OPE_MI	MI Accum RE after Acquisition (calc)	Y	N	N						1
		MI_Exp	SB Method MI Expense (calc)	Y	N	N						1
		EQ_Pickup	Equity Pickup (calc)	Y	N	N						1
		Override_Elim	Override Elimination (calc)	Y	N	N						1
		GP_Elim	Gross Profit Elimination	Y	N	N						1
		LedgerAdjust	Total After Ledger Journals	N	N	N						1
		USGAAP	US GAAP adjustments	N	N	N						1
		IFRS	IFRS adjustments	N	N	N						1
		HFMJE	HFM Only JE	N	N	N						1
		LOCJE	JFM JE to Post in Local Ledger	N	N	N						1
		[None]	FDM Ledger loads	N	N	N						1

Custom 4 Data Type

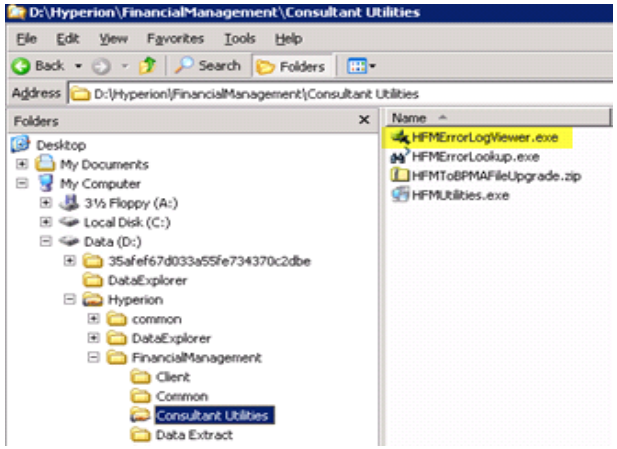
- Use to facilitate reporting of various types of data collected. For example: fed from ledger versus different types of journals
- Benefits
  - Allows for reporting of various data types in reports and smart view files
  - Can easily add another data type as needed
  - Can apply custom attributes to data type. For example: calculate, UD fields, etc.

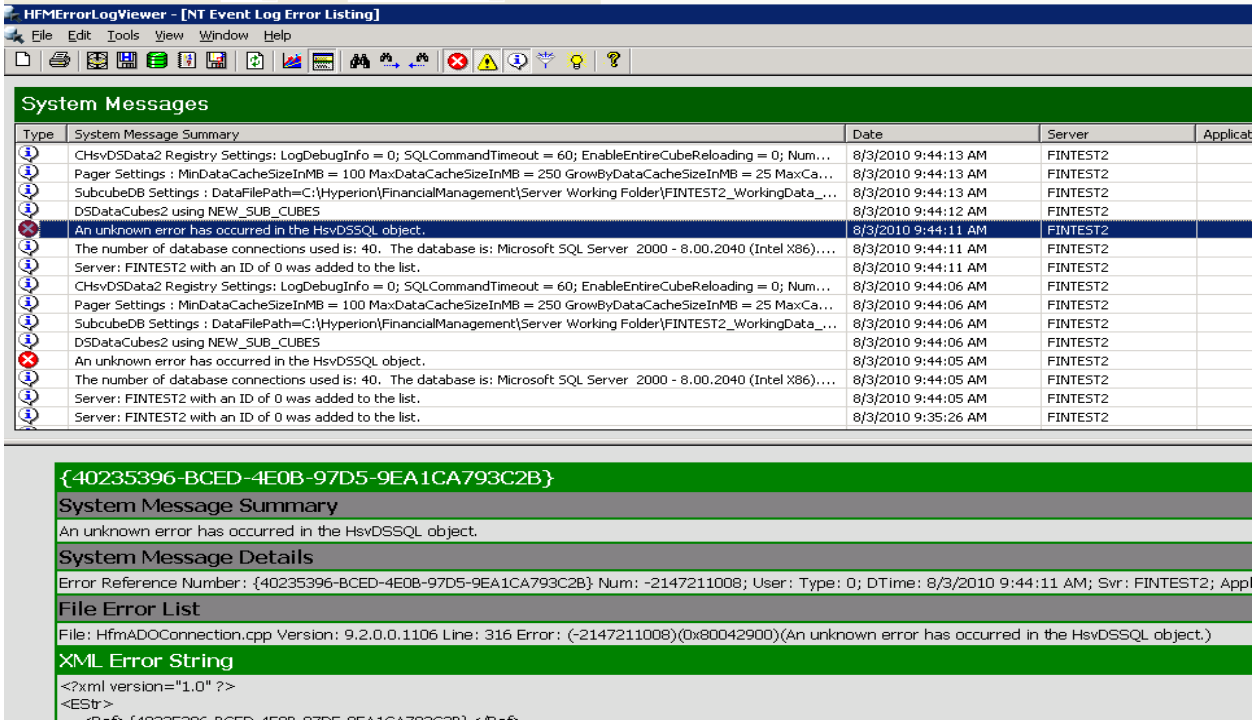
? ? ? ? ? ?

Error Reference Number: {AE04ED18-50C6-4135-B84D-09411E8B405B}

Decode admin messages

- Using the Consulting Tool – **HFMErrorLogViewer.exe**





Decode admin messages

{D7D6794F-19FC-418C-8D1B-EF83390CA6B1}

System Message Summary

An unknown error has occurred in the HsvDSSQL object.

System Message Details

Error Reference Number: {D7D6794F-19FC-418C-8D1B-EF83390CA6B1} Num: -2147211008; User: Type: 0; DTime: 7/24/2010 5:25:15 PM; Svr: FINTEST2; Application:

File Error List

File: HfmADOConnection.cpp Version: 9.2.0.0.1106 Line: 316 Error: (-2147211008)(0x80042900)(An unknown error has occured in the HsvDSSQL object.)

XML Error String

```
<?xml version="1.0" ?>
<EStr>
  <Ref>{D7D6794F-19FC-418C-8D1B-EF83390CA6B1}</Ref>
  <User></User>
  <DBUpdate>46202048</DBUpdate>
  <ESec>
    <Num HEX=0x80042900>-2147211008</Num>
    <Description Src=HFM>An unknown error has occured in the HsvDSSQL object.</Description>
    <Type>1</Type>
    <DTime>7/24/2010 5:25:15 PM</DTime>
    <Svr>FINTEST2</Svr>
    <File>HfmADOConnection.cpp</File>
    <Line>316</Line>
    <Ver>9.2.0.0.1106</Ver>
    <DStr></DStr>
    <ExEn></ExEn>
  </ESec>
</EStr>
```


Decode admin messages

{A85E4FC5-015F-4573-A925-A6867F240041}

System Message Summary

The number of database connections used is: 40. The database is: Microsoft SQL Server 2000 - 8.00.2040 (Intel X86). Multi-server is not ON

System Message Details

Error Reference Number: {A85E4FC5-015F-4573-A925-A6867F240041} Num: 0; User: Type: 0; DTime: 7/26/2010 7:45:39 AM; Svr: FINTEST2; Application:

File Error List

File: CHsvDSSQL.cpp Version: 9.2.0.0.1106 Line: 398 Error: (0)(0x0)(No error)

XML Error String

```
<?xml version="1.0" ?>
<EStr>
  <Ref>{A85E4FC5-015F-4573-A925-A6867F240041}</Ref>
  <User></User>
  <DBUpdate>43798696</DBUpdate>
  <ESec>
    <Num HEX=0x0>0</Num>
    <Description Src=None>No error</Description>
    <Type>1</Type>
    <DTime>7/26/2010 7:45:39 AM</DTime>
    <Svr>FINTEST2</Svr>
    <File>CHsvDSSQL.cpp</File>
    <Line>398</Line>
    <Ver>9.2.0.0.1106</Ver>
    <DStr>The number of database connections used is: 40. The database is: Microsoft SQL Server 2000 - 8.00.2040 (Intel X86). Multi-server is not ON</DStr>
    <ExEn></ExEn>
  </ESec>
</EStr>
```

Message: "There were not enough connections in the SQL Connection Pool. The number of connections is 40. The Thread ID is 5700. Please consider increasing the SQL Connection Pool size." [ID 597517.1]

Modified 28-JAN-2010 *Type* PROBLEM *Status* PUBLISHED

In this Document

Applies to:

Hyperion Financial Management - Version: 3.5.0.0.00 to 11.1.1.3.00 - Release: 3.5 to 11.1
z*OBSOLETE: Microsoft Windows

Symptoms

Message: "There were not enough connections in the SQL Connection Pool. The number of connections is 40. The Thread ID is 5700. Please consider increasing the SQL Connection Pool size."

Cause

User activity is high enough to use all existing pooled connections.

Solution

Refer to the Installation guide section on *Database Connection Pooling* for details on how connections are used.

It is not necessary to increase this setting unless there are many messages being logged. An occasional message might not mean that changing the parameter is required.

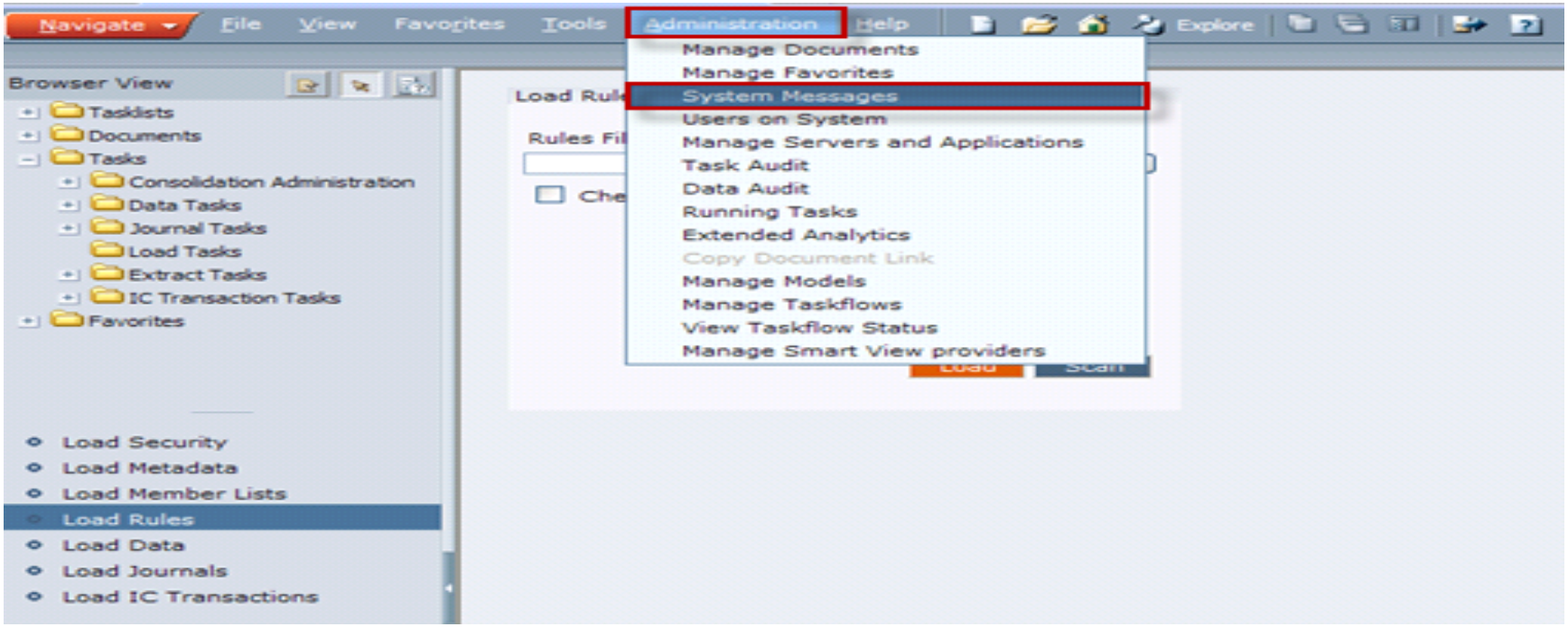
The configured value can be changed in different ways depending on the version of Financial Management.

11.1.1 and above:

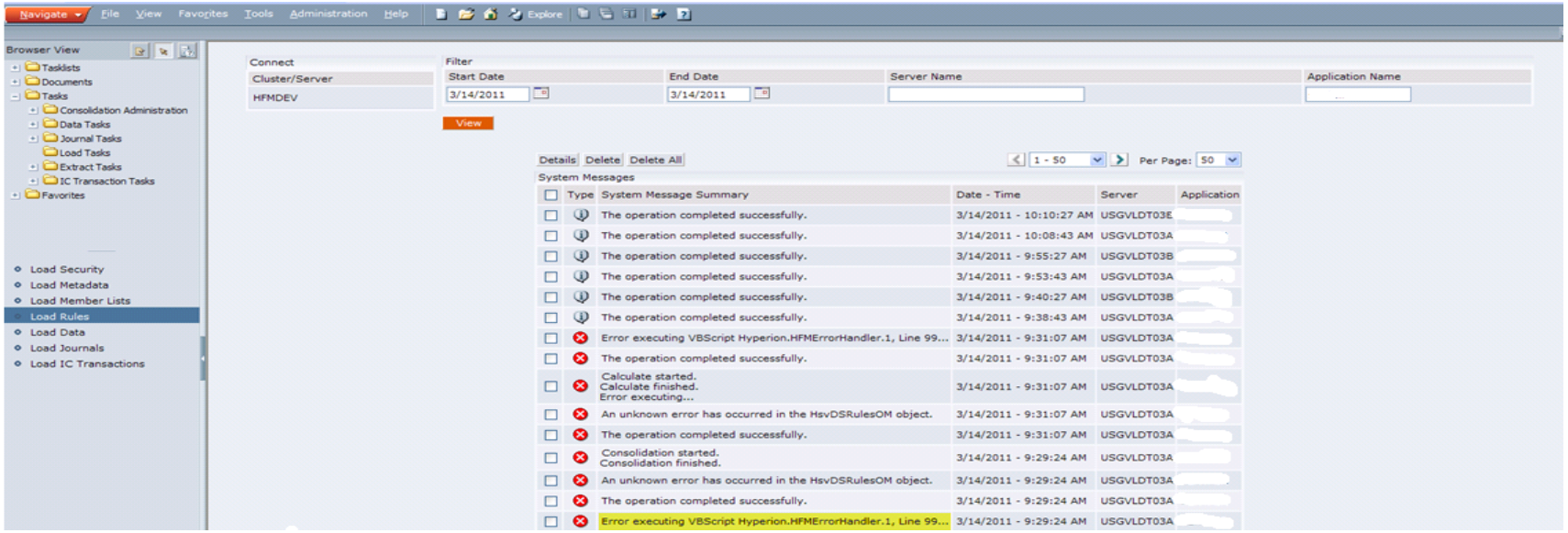
1. run the EPM Configuration Utility;
2. expand *Financial Management Configuration* and select Configure Database. Make sure no other task is selected.
3. Make sure the basic information is correct, and click on *Advanced Configuration*, to modify the parameter.
4. Follow the wizard and click Finish when completed.

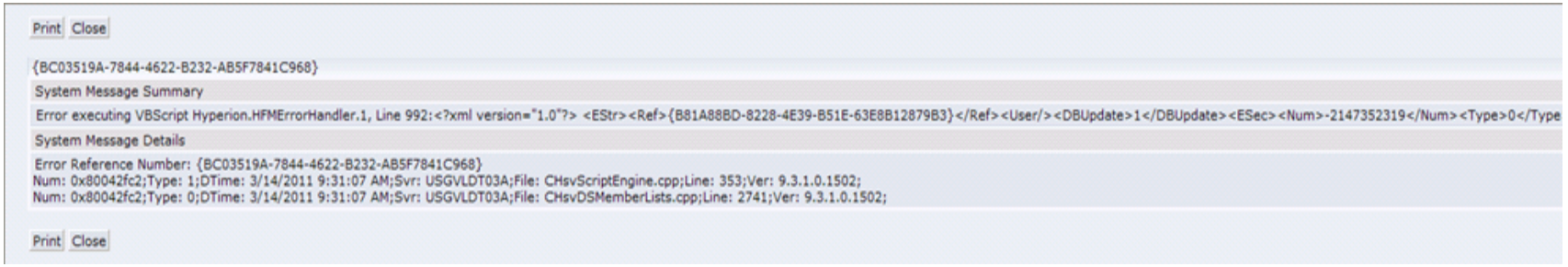
Decode admin messages

View error log via the web

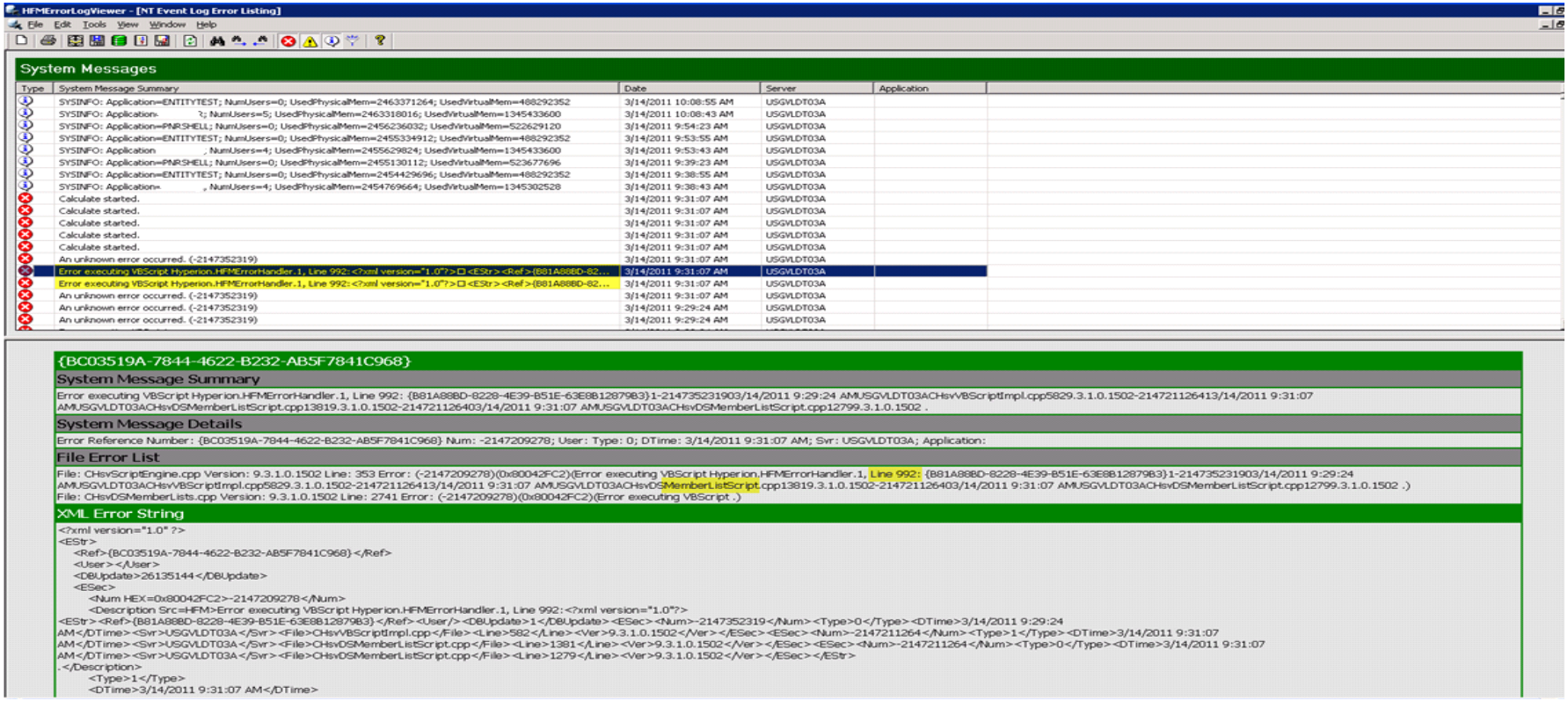


Decode admin messages





Decode admin messages



How can rules save time for your admin and reduce your data risk?

Rules to Perform Common Maintenance

- Use rules to copy data from one scenario to another to facilitate FX analysis

```
If POVValue = "USD Adjs" or (Is_Base and (POVValue = "<Entity Currency>" or POVValue = "<Entity Curr Adjs>")) Then

    'Loop through each base level accounts
    For Each Acc In HS.Account.List("","[Base]")

        'If the account is not one of the Global rate accounts
        If Acc <> "AVGRATE" And Acc <> "EOMRATE"  Then

            '-------------------------------------------------------------------------------------------------------
            'Copy ACTUAL data into the ActatPlan rate scenario
            If POVScen = "ActatPlan" Then

                'Clear the destination
                Hs.Clear "A#" & Acc

                'Copy the data
                Hs.Exp "A#" & Acc & "=A#" & Acc & ".W#Periodic.S#Actual"

            'Copy ACTUAL data into the ActatPriorYr rate scenario
            ElseIf POVScen = "ActatPriorYr" Then

                'Clear the destination
                Hs.Clear "A#" & Acc

                'Copy the data
                Hs.Exp "A#" & Acc & "=A#" & Acc & ".S#Actual"
```


Rules to Perform Common Maintenance

- Use rules to copy the exchange rates from one scenario to another

```
If StrComp(POVEnt, "[None]", vbTextCompare) = 0 Then

    'Loop through each base level accounts
    For Each Acc In HS.Account.List("","[Base]")

        'If the scenario is ActAtPriorYr then pull the rates from the Prior Year in the ACTUAL scenario
        If StrComp(POVScen, "ActatPriorYr", vbTextCompare) = 0 Then

            'Copy the rates
            Hs.Exp "A#" & Acc & "=A#" & Acc & ".S#Actual.Y#Prior"


        'If the scenario is ActAtPlan then pull the rates from the PLAN scenario
        ElseIf StrComp(POVScen, "ActatPlan", vbTextCompare) = 0 Then

            'Copy the rates
            Hs.Exp "A#" & Acc & "=A#" & Acc & ".S#Plan"
```


Rules to Perform Common Maintenance

- Use a rule to change the status of flexing scenarios to Impacted so that users will know consolidation is required

```
'-------------------------------------------------------------------------------------------------------
'Impact Flexing Scenarios
'-------------------------------------------------------------------------------------------------------
If POVValue = "USD Adjs" or (Is_Base and (POVValue = "<Entity Currency>" or POVValue = "<Entity Curr Adjs>")) Then
    If HS.Scenario.Member = "Actual" Then

        HS.ImpactStatus "P#Cur.S#ActatPlan"
        HS.ImpactStatus "P#Cur.S#ActatPriorYr"

    ElseIf HS.Scenario.Member = "Pre_Plan" Then

        HS.ImpactStatus "P#Cur.S#Pre_PlanAtPriorYr"

    ElseIf HS.Scenario.Member = "CurrForecast" Then

        HS.ImpactStatus "P#Cur.S#CurrFcstAtPriorYr"

    End If 'POVScen
End If 'POVValue = "USD Adjs" or (Is_Base and (POVValue = "<Entity Currency>" or POVValue = "<Entity Curr Adjs>"))
```


Rules to Perform Common Maintenance

- Use NoInput rule to prevent users from making changes to your flexing scenarios

```
With HS

    '*****************************************************
    'No input for specific scenarios

    For Each Acc In .Account.List("", "[Base]")

        .NoInput "A#" & Acc & ".S#ActatPriorYr"
        .NoInput "A#" & Acc & ".S#ActatPlan"
        .NoInput "A#" & Acc & ".S#Pre_PlanAtPriorYr"
        .NoInput "A#" & Acc & ".S#CurrFcstAtPriorYr"

    Next 'For each Acc
```


Rules to Perform Common Maintenance

- Block input to parent currency adjustments (PCA)

- Using PCA's when you have multiple hierarchies can prove difficult when trying to reconcile between hierarchies

```
With HS

    '*************************************************************************************************
    'No input after 2010 on <PCA> (journals must be in local currency); [PA] will never be usable
    '(as AllowAdjFromChildren always set to No in the metadata)

    For Each Yea In Array("2011", "2012", "2013", "2014", "2015")

        aLi = .Entity.List("", "")
        For i = LBound(aLi) to UBound(aLi)

            .NoInput "E#" & Ent & ".V#<Parent Curr Adjs>.Y#" & Yea

        Next 'Entity Loop
    Next 'Year in Array Loop

End With
```


Rules to Perform Common Maintenance

- NoInput rules can be useful in other areas if you want to add a time stamp on using specific ICP's for example

```
For Each Yea In Array("2011", "2012", "2013", "2014", "2015")

    HS.NoInput "I#ACUTROL.Y#" &Yea
    HS.NoInput "I#AFLTOTUS.Y#" &Yea
    HS.NoInput "I#APAC_JV_HK_ADJ.Y#" &Yea
    HS.NoInput "I#APAC_JV_RMB_ADJ.Y#" &Yea
    HS.NoInput "I#AQUACULTURE.Y#" &Yea
    HS.NoInput "I#AQUARIUM.Y#" &Yea
    HS.NoInput "I#ASIATOTU.Y#" &Yea

Next 'Year in Array Loop
```


Business Rules – Tip of the Day

- Use Notepad++ to view and edit your rules file

- Select Language>V>VB for optimal viewing of rules

```
'*************************************************************************
'                       Balance Sheet Difference Function
'*************************************************************************
Function BS_Diff (Acct)

    '*********************************************************************
    'A function routine that calculates the difference between Open Balance of an
    ' Asset or Liability tagged Account. Will be used for Cash Flow calculations.

    If UCase(HS.Account.AccountType(Acct)) = "ASSET" Then
        BS_Diff = "(A#" & Acct & ".I#[ICP Top].C1#DEPTS.C3#PRODTP.C4#TOT_DataType.P#Last.Y#Prior" & _
            " - A#" & Acct & ".I#[ICP Top].C1#DEPTS.C3#PRODTP.C4#TOT_DataType)"
    Else
        BS_Diff = "(A#" & Acct & ".I#[ICP Top].C1#DEPTS.C3#PRODTP.C4#TOT_DataType" & _
            " - A#" & Acct & ".I#[ICP Top].C1#DEPTS.C3#PRODTP.C4#TOT_DataType.P#Last.Y#Prior)"
    End If 'If UCase
    '*********************************************************************

End Function ' BS_Diff (Acct)
```


Anyone enjoy working on the Cash Flow?

Cash Flow Option #1

- Dynamic Cash Flow available at base entity level
- Accomplished by using the below 3 account groupings
  - Balance Sheet accounts, at selected levels
  - Balance Sheet roll-forward accounts to match selected levels
  - Cash Flow accounts
- Additional items used in HFM
  - Account lists
  - User Defined field
  - Custom dimensions
  - Business Rules

Cash Flow Option #1

- Benefits
  - Cash Flow reporting available at base entity level
  - Does not require rule update with metadata changes
  - Keeps your custom dimension open on the balance sheet for other uses
  - Easy reporting with accounts in rows and movements in columns
  - Allows for CF adjustments (non-cash items, acquisitions)

Dynamic Cash Flow at base entity

Steps:

1) Determine what account level on the balance sheet you want to track account balance changes

	Description	ConsolidationAccountType
ARNet	Trade Receivables Net	Asset
ARGross	Gross Receivables	Asset
B0050	B0050 - A/R	Asset
B0055	B0055 - A/R Rebill	Asset
ARBadDebt	Bad Debt Reserve	Asset
B0060	B0060 - Doubtful Acct Allowance	Asset
B0065	B0065 - Bad Debt Write Off	Asset
B0070	B0070 - Bad Debt Recoveries	Asset

Dynamic Cash Flow at base entity

2) Create a group of Balance Sheet roll forward accounts that match the accounts where you plan to track cash flow

- These will be used to track movements for these accounts

Format	Description	ConsolidationAccountType	C1	C2	C3	C4	User defined1	User defined 2	User defined 3
BSRfwdACCT	Balance Sheet Roll-forward Accounts	GroupLabel							
BS_Rfwd	Balance Sheet Roll-forward	Asset		Product	TOT_BS_Rfwd	DataType			
CashRoll	Cash Roll-forward	Asset			TOT_BS_Rfwd	DataType			CASH
ARNetRoll	AR Roll-forward	Asset			TOT_BS_Rfwd	DataType			ARNET
InventoryRoll	Inventory Roll-forward	Asset		Product	TOT_BS_Rfwd	DataType			INVENTORY
OthCurrAssetsRoll	Other Current Assets Roll-forward	Asset			TOT_BS_Rfwd	DataType			OTHCURRASSETS
DefTaxRoll	Deferred Income Tax Roll-forward	Asset			TOT_BS_Rfwd	DataType			
STDefTaxAssetRoll	ST Deferred Tax Asset Roll-forward	Asset			TOT_BS_Rfwd	DataType			DEFTAXSTASS
LTDefTaxAssetRoll	LT Deferred Tax Asset Roll-forward	Asset			TOT_BS_Rfwd	DataType			DEFTAXLTASS
LTDefTaxLiabRoll	LT Deferred Tax Liability Roll-forward	Liability			TOT_BS_Rfwd	DataType			DEFTAXLTLIAB

Dynamic Cash Flow at base entity

3) Determine various movements for changes in your Balance Sheet balances. For example: additions, disposals, FX or just a change in the balance.

- These will be assigned to the related movement account as a C2.

Members in Tiered Format	Alias
TOT_BS_Rfwd	Total Balance Sheet Roll-forward
RFOpen	Opening balance
RFBB	Beginning Balance Adj
RFChg	Account change/movement
RFFX	FX difference

Dynamic Cash Flow at base entity

4) Assign the same user defined of ARNET to both the Balance Sheet account and related roll-forward account

- This will be used in the business rule to calculate the change

	Description	ConsolidationAccountType	User defined1	User defined 2	User defined 3
ARNet	Trade Receivables Net	Asset		ARNET	
ARGross	Gross Receivables	Asset			
B0050	B0050 - A/R	Asset			
B0055	B0055 - A/R Rebill	Asset			
ARBadDebt	Bad Debt Reserve	Asset			
B0060	B0060 - Doubtful Acct Allowance	Asset			
B0065	B0065 - Bad Debt Write Off	Asset			
B0070	B0070 - Bad Debt Recoveries	Asset			
BSRfwdACCT	Balance Sheet Roll-forward Accounts	GroupLabel			
BS_Rfwd	Balance Sheet Roll-forward	Asset			
CashRoll	Cash Roll-forward	Asset			CASH
ARNetRoll	AR Roll-forward	Asset			ARNET
InventoryRoll	Inventory Roll-forward	Asset			INVENTORY
OthCurrAssetsRoll	Other Current Assets Roll-forward	Asset			OTHCURRASSETS
DefTaxRoll	Deferred Income Tax Roll-forward	Asset			
STDefTaxAssetR	ST Deferred Tax Asset Roll-forward	Asset			DEFTAXSTASS
LTDefTaxAssetR	LT Deferred Tax Asset Roll-forward	Asset			DEFTAXLTASS
LTDefTaxLiabRol	LT Deferred Tax Liability Roll-forward	Liability			DEFTAXLTLIAB

Dynamic Cash Flow at base entity

5) Assign C3 of TOT_CF to all your Cash Flow accounts

- This will allow for various adjustments to the Cash Flow accts

Format	Description	ConsolidationAccountType	C1	C2	C3	C4
CFOperItems	Changes in current operating items	Expense			TOT_CF	
CFAR	Trade receivables	Expense			TOT_CF	
CFInventory	Inventories	Expense		Product	TOT_CF	
CFAP	Accounts payable	Expense			TOT_CF	
CFIC	Inter-company	Expense			TOT_CF	
CFAccLiab	Accrued expenses	Expense		Product	TOT_CF	
CFIncTax	Income tax payable/receivable	Expense			TOT_CF	
CFOtherCA	Prepaid and others net	Expense			TOT_CF	
CFDiscOps	Net cash provided/used by discontinued operations	Expense			TOT_CF	

Members in Tier	Alias
TOT_CF	Total Cash Flow
CFCalc	Calculate Cash Flows
CFAdj	Adjust for non cash items via HFM JE's
CFAcq	Reclass for acquisition

```
' !Trade receivables
    HS.Exp "A#CFAR.C3#CFCALC = " & BS_Diff ("ARNet")
```


Cash Flow Option #2

- Cash Flow entered at consolidated entity level
- Accomplished using the below HFM items:
  - Cash Flow accounts
  - Account member list
  - Business rules

Cash Flow Option #2

- Benefits
  - Allows for easy input at consolidated financial statement level
  - No rule changes required as new accounts are added to Cash Flow statement
  - Easy to add additional accounts for input
  - Very flexible

Cash Flow Option #2

1) Determine if you are able to build a Cash Flow at the base entity level. If not, then determine what Cash Flow accounts should be input at your consolidated level.

- Assign a user defined field to those accounts.
- Set up dynamic account member list, to be used in the rule.

Member	Description	AccountType	User defined1
C100000	Net earnings	Revenue	Consol_Input
C100050	Depreciation and amortization	Revenue	Consol_Input
C100100	After-tax earnings from joint ventures	Revenue	Consol_Input
C100150	Stock-based compensation	Revenue	Consol_Input
C100200	Deferred income taxes	Revenue	Consol_Input
C100250	Distribution of earnings from joint venture	Revenue	Consol_Input
C100300	Tax benefit on exercised options	Revenue	Consol_Input
C100350	Pension, other postretirement, and postemployment benefit costs	Revenue	Consol_Input
C100400	Restructuring, impairment and other exit costs (income)	Revenue	Consol_Input
C100450	Gain on insurance settlement	Revenue	Consol_Input
C100900	Receivables	Revenue	Consol_Input
C100910	Inventories	Revenue	Consol_Input
C100920	Prepaid expenses and other current assets	Revenue	Consol_Input
C100930	Accounts payable	Revenue	Consol_Input
C100940	Other Current Liabilities	Revenue	Consol_Input

Cash Flow Option #2

- In business rules use the HS.Input statement with your account list to define those accounts that should be input at the consolidated entity level.

```
'Rules in effect for the next 5 years - update rules for future years, once we get to 2014
For Each Yea In Array("2010", "2011", "2012", "2013", "2014")

    'Allow input into the Consol_Input account list
    For Each Acc In .Account.List("", "Consol_Input")

        .Input "Y#" & Yea & ".A#" & Acc

    Next 'For each Acc In .Account.List("", "Consol_Input"

Next 'For each Yea
```


eCapital Advisors
A MANAGEMENT CONSULTING FIRM