Optimize SM Application

June 14, 2021

JM – all dashboards, repots, charts in SM, Review with me and Make sure these are working

GA – what is the purpose of Services menu

GA-we may have many unwanted features in menu/feature table – why not delete them too.

GA-if features are not in UI but WS exists how can one find ws is use for an unwanted featrure ?

Alternately we can identify methods in use for current features in UI and go that way.

GA: Combine DCIM and Visio Stencils menu to All Accounts; rename All Accounts to “Accounts”

GA: Clicking on Home menu for SM shows features of RMS menu

# 1 Review complete SM to determine unwanted features

Our goal is to delete unwanted features permanently

Delete Following feature from SM UI

**Application QA**

AppQA: Critical Alerts

AppQA: Logout

**Menu: MyTask**

My Task/My Task/User Settings/Alerts

My Task/My Task/User Settings/VOIP Call Log

**Menu: All Accounts, DCIM and Visio Stencils**

Menu: Remove following features from menu: All Accounts, DCIM and Visio Stencils, New Sales

All Accounts/Select Accounts/By Purchase/Professional Only ;combines results with enterprise

All Accounts/Select Accounts/By Purchase/Basic Only ; ;combines results with enterprise

All Accounts/Select Accounts/By Company Size/Small

All Accounts/Select Accounts/By Company Size/Medium

All Accounts/Select Accounts/By Company Size/Large

All Accounts/Select Accounts/By Company Size/Others

All Accounts/Select Accounts/By Company Size/Competition ;

All Accounts/Select Accounts/By Company Size/All Accounts ;

Administration/Services/Catalog/General Supplies

Administration/Services/Catalog/Office Assets

Test application that nothing else is broken.

# GA: How to deliver updated SM code and database

# 2 Optimize SM Application

Idea:

**How to store a TraceMethod parameter that can be accessed from WS as awell as within Database SP\_?**

Step - 1

## //Modify SM Database

Add a Parameter to table: zInfo. Trace\_WS\_Methods = 0 or 1

Step - 2

## //Modify SM Database

**//Add a table in SM database**

**Table: TraceMethods (Fields: flag int, <Name> nvarchar(100), LastUpdated(Date) )**

Where Flag:int 0|1|2|9; [0 for WS\_MethodName, 1 for SP\_Proc, 2 for Funcname, 9 for Table names]

Step - 3

**//Add a SP\_ TraceMthod(flag, String(<Coma separated names** of one of {Method/SP\_/Func/Table} **>))**

{

Load Trace\_WS\_Methods = 1 from zinfo

If (Trace\_WS\_Methods == 1)

Then Append(flag, Array intp (String\_Split((<Coma separated names>)) to table: **TraceMthod**

}

Step - 4

//modify each SP\_ in each Programmability object of SM/RMS by adding these 3 calls as first lines

Call **SP\_ TraceMthod(1, <Coma separated SP\_names>)**

Call **SP\_ TraceMthod(2, <Coma separated Func\_names>)**

Call **SP\_ TraceMthod(9, <Coma separated Table\_names>)**

Step - 5

## //Modify SM/RMS WebServices

//Add method Develop a C# code (WS\_method)

WS\_TraceMthod(<Coma separated Method names>)

{

Call SP\_ TraceMthod(0, <Coma separated Method names>)

}

Step - 6

**Call method: WS\_TraceMthod(0, <Coma separated Method names>)**

as a first call in each ws\_method of our SM and RMS webservices.

IMPORTANT:

If you missed calling TraceMethod in a Method/procedure you can accidently delete it requiring unnecessary troubleshooting.

Step – 7

Purge content of table: **TraceMthod**

**//Test each and every feature of SM and RMS**

//Use every feature with all options, including all reports etc. (No exception)

Step - 8

## //When all testing done, query what can be deleted

WS\_Method\_Deleted [ ] = Select Distinct WS\_Methods from **TraceMthod**

SP\_Delete[ ] = Select Distinct SP from **TraceMthod**

Func\_Deleted [ ] = Select Distinct Func from **TraceMthod**

Table\_Deleted [ ] = Select Distinct Tables from **TraceMthod**

**Note: Shape these lists with NZ developrs for comments if any.**

Step – 9a

**// purge WS methods and Develop SQL script to purge SM Database objects in following order**

1. Edit WS projects and Delete WS\_method\_Deleted[ ] and any unused variables if any. Recompile WS projects
2. Script to delete SP\_Delete[ ]
3. Script to delete Func\_Delete[ ]
4. Script to delete Table\_Delete[ ]

If any errors encountered, then troubleshoot; likely you deleted something you should not have deleted.

Step – 9b

Modify Script in Step 9a, as needed

Step-10

Repeat Step 7 and Step 8; and Step 9b

until

There is not Distinct records produced in Step 8

All SM features are working as before optimizing

Step-11

Set zInfo. Trace\_WS\_Methods = 0

//Clean up WS\_methods

Delete WS\_TraceMethod we defined in Step-5, also delete it from within each WS\_method call

//we can leaver trace method in Database for future use.

Now we are left with WS methods we require as well as DB objects we need for SM/RMS application

# 3 Performance optimization

Step -12

Divide remaining Features between NZ team and Mihir’s team as follows

<To be developed by Y>

Step-13

Optimize database code for required features

Note: we will not modify WS\_Methods

Do for each feature of the SM/RMS

Find out the purpose of each SP and function and also table schema in use (in case there are unwanted fields, index missing, NULL value handling, OR a field if added may increase performance, a trigger missing that will fix an issue etc.

Note: Discuss with NZ developers if needed to fully understand.

Make changes to (SP and function and also table schema) as needed to improve performance.

Verify if WS\_method still working as expected.

Now Test the respective Feature completely

Step-14

Performance Script - Extract script that will drop old tables, code etc and create new for all changes made

We will call it Performance script

Deliverables

**Purged WS\_Project**

**Purge Script** – purge unwanted Database objects in SM

**Performance Script** - Drop and Create new Func, SP to improve performance

Tips for SQL developer

As a default use, users donot use any filters and hence such SP\_ should be created smart or separately to handle default behavior of application.

**//Automate some of the tasks.**

Use following queries to list SP\_ func and Table names

We can join against what goes in Trace Table

And auto develop script to drop these objectshttps://docs.microsoft.com/en-us/sql/relational-databases/user-defined-functions/view-user-defined-functions?view=sql-server-ver15

**How to Get SP\_ names**

SELECT

SCHEMA\_NAME(schema\_id) AS [Schema], name FROM sys.objects WHERE type\_desc = 'SQL\_STORED\_PROCEDURE';

SELECT name FROM sys.objects WHERE type\_desc = 'SQL\_STORED\_PROCEDURE';

**Iterate thru each name to get SP\_ definitions**

SELECT definition

FROM sys.sql\_modules

WHERE object\_id = object\_id(**'uspGetAlbumsByArtist'**);

**How to Get Func**

SELECT name, definition, type\_desc

FROM sys.sql\_modules m

INNER JOIN sys.objects o

ON m.object\_id=o.object\_id

WHERE type\_desc like '%function%'

**How to get Tables**

SELECT \* FROM SYSOBJECTS where type ='U'

How to remove unused references from VS code

