ECM Library Web Services API

Technical Usage Manual

Contents

[The purpose of the ECM API 5](#_Toc246917254)

[What is lost by searching thru a service and not thru our client? 5](#_Toc246917255)

[ECM Web Services Calls: 6](#_Toc246917256)

[Installing a reference to the service 6](#_Toc246917257)

[The Details of the Web Services 8](#_Toc246917258)

[Exposed Web Services 8](#_Toc246917259)

[ECM Service Execution Parameters: 8](#_Toc246917260)

[Mandatory Search Parameters 8](#_Toc246917261)

['\*\* Operational Data 8](#_Toc246917262)

[Search Data 8](#_Toc246917263)

[Database Class Parameters 8](#_Toc246917264)

[EMAIL Search Parameters 8](#_Toc246917265)

[Content Search Parameters 8](#_Toc246917266)

[Programming Examples: 8](#_Toc246917267)

[Search for a document by create date between two dates 8](#_Toc246917268)

[Search for a document by create date before a date 8](#_Toc246917269)

[Search for a document by create date ON a specific date 8](#_Toc246917270)

[Execute a Dynamic SQL Statement 8](#_Toc246917271)

[Restore an EMAIL 8](#_Toc246917272)

[Search for a document created after a specific date 8](#_Toc246917273)

[Search for a document created between dates 8](#_Toc246917274)

[Search for Email after a specific SENT date 8](#_Toc246917275)

[Search for Email after a specific RECEIVED date 8](#_Toc246917276)

[Endpoint configuration error: 8](#_Toc246917277)

[ECM Web Services Source Code: 8](#_Toc246917278)

[<WebMethod()> Public Function getConnectionString() As String 8](#_Toc246917279)

[<WebMethod()> Public Function genDocSearchSql(ByVal SessionGuid$, ByVal SearchString$) As String 8](#_Toc246917280)

[<WebMethod()> Public Function genEmailSearchSql(ByVal SessionGuid$, ByVal SearchString$) As String 8](#_Toc246917281)

[<WebMethod()> Public Function SearchEmail(ByVal SessionGuid$, ByVal SearchString$) As DataSet 8](#_Toc246917282)

[<WebMethod()> Public Function SearchEmailXML(ByVal SessionGuid$, ByVal SearchString$) As String 8](#_Toc246917283)

[<WebMethod()> Public Function SearchDocuments(ByVal SessionGuid$, ByVal SearchString$) As DataSet 8](#_Toc246917284)

[<WebMethod()> Public Function SearchDocumentsXML(ByVal SessionGuid$, ByVal SearchString$) As String 8](#_Toc246917285)

[<WebMethod()> Public Function setConnStr(ByVal SessionGuid$, ByVal ConnStr$) As Boolean 8](#_Toc246917286)

[<WebMethod()> Private Function setSession(ByVal SessionVarName$, ByVal tVar$) As String 8](#_Toc246917287)

[<WebMethod()> Private Function getConnStr() As String 8](#_Toc246917288)

[<WebMethod()> Private Function iCount(ByVal SelectSql As String, ByVal ConnStr$) As Integer 8](#_Toc246917289)

[<WebMethod()> Public Function CountSearchTokens(ByVal SearchString As String) As String 8](#_Toc246917290)

[<WebMethod()> Public Function getSqlDS(ByVal SelectSql As String, ByVal ConnStr$) As DataSet 8](#_Toc246917291)

[<WebMethod()> Public Function getDataSetCount(ByVal SelectSql As String, ByVal ConnStr$) As Integer 8](#_Toc246917292)

[<WebMethod()> Public Function getSqlDsXML(ByVal SelectSql As String, ByVal ConnStr$) As String 8](#_Toc246917293)

[<WebMethod()> Public Function iCountRecordsInTable(ByVal Tbl As String) As Integer 8](#_Toc246917294)

[<WebMethod()> Public Function getSourceLength(ByVal SourceGuid$) As Double 8](#_Toc246917295)

[<WebMethod()> Public Function getSourceImage(ByVal SourceGuid As String, ByVal SourceLength As Integer) As Byte() 8](#_Toc246917296)

[<WebMethod()> Public Function getEmailImage(ByVal EmailGuid$, ByVal SourceLength As Integer) As Byte() 8](#_Toc246917297)

[<WebMethod()> Public Function restoreDocument(ByVal SessionGuid$, ByVal SourceGuid$, ByVal SavePath$, ByVal OverWrite As Boolean) As Boolean 8](#_Toc246917298)

[<WebMethod()> Public Function restoreEmail(ByVal SessionGuid$, ByVal EmailGuid$, ByVal SavePath$, ByVal OverWrite As Boolean) As Boolean 8](#_Toc246917299)

[<WebMethod()> Public Function getEmailAttachmentImage(ByVal SessionGuid$, ByVal ContentType As String, ByRef SourceLength As Integer, ByVal AttachmentID As Integer) As Byte() 8](#_Toc246917300)

['<WebMethod()> 'Public Function WriteToWebSvcLog(ByVal Msg$) As Boolean 8](#_Toc246917301)

['<WebMethod()> 'Sub ResolveName(ByVal name As String) 8](#_Toc246917302)

[<WebMethod()> Private Function getUserIP() As String 8](#_Toc246917303)

[<WebMethod()> Private Function MachineRegistered() As Boolean 8](#_Toc246917304)

[<WebMethod()> Private Function GetHostName() As String 8](#_Toc246917305)

[<WebMethod()> Private Function GetMachineIPAddr(ByVal SessionGuid$) As String 8](#_Toc246917306)

[<WebMethod()> Public Function GetClientIPAddr(ByVal SessionGuid$) As String 8](#_Toc246917307)

[<WebMethod()> Private Sub updateIPAddr(ByVal SessionGuid$, ByVal UpdateCode As Integer) 8](#_Toc246917308)

[<WebMethod()> Public Sub setSearchString(ByVal SessionGuid$, ByVal SearchString As String) 8](#_Toc246917309)

[<WebMethod()> Public Function getSearchString(ByVal SessionGuid$) As String 8](#_Toc246917310)

[<WebMethod()> Public Sub setEmailIncludeAttachments(ByVal IncludeAttachments As Boolean) 8](#_Toc246917311)

[<WebMethod()> Public Sub setIsAdmin(ByVal SessionGuid$, ByVal IsAdmin As Boolean) 8](#_Toc246917312)

[<WebMethod()> Public Function getIsAdmin(ByVal SessionGuid$) As Boolean 8](#_Toc246917313)

[<WebMethod()> Public Sub setIsGlobalSearcher(ByVal SessionGuid$, ByVal IsGlobalSearcher As Boolean) 8](#_Toc246917314)

[<WebMethod()> Public Function getIsGlobalSearcher(ByVal SessionGuid$) As Boolean 8](#_Toc246917315)

[<WebMethod()> Public Sub setCurrUserGuidID(ByVal SessionGuid$, ByVal vCurrUserGuidID As String) 8](#_Toc246917316)

[<WebMethod()> Public Sub setMasterContentOnly(ByVal SessionGuid$, ByVal vMasterContentOnly As Boolean) 8](#_Toc246917317)

[<WebMethod()> Public Sub setMyContentOnly(ByVal SessionGuid$, ByVal vMyContentOnly As Boolean) 8](#_Toc246917318)

[<WebMethod()> Public Sub setNbrOfErrors(ByVal SessionGuid$, ByVal vNbrOfErrors As Integer) 8](#_Toc246917319)

[<WebMethod()> Public Sub setSql(ByVal SessionGuid$, ByVal vSql As String) 8](#_Toc246917320)

[<WebMethod()> Public Sub setErrMsg(ByVal SessionGuid$, ByVal vErrMsg As String) 8](#_Toc246917321)

[<WebMethod()> Public Sub setMinRating(ByVal SessionGuid$, ByVal vMinRating As Integer) 8](#_Toc246917322)

[<WebMethod()> Public Sub setClipBoardData(ByVal SessionGuid$, ByVal vClipBoardData As String) 8](#_Toc246917323)

[<WebMethod()> Public Sub getClipBoardData(ByVal SessionGuid$, ByVal vClipBoardData As String) 8](#_Toc246917324)

[<WebMethod()> Public Sub setCkLimitToLib(ByVal SessionGuid$, ByVal vCkLimitToLib As Boolean) 8](#_Toc246917325)

[<WebMethod()> Public Sub setLibrary(ByVal SessionGuid$, ByVal vCbLibrary As String) 8](#_Toc246917326)

[<WebMethod()> Public Sub setEmailSubject(ByVal SessionGuid$, ByVal vTxtSubject As String) 8](#_Toc246917327)

[<WebMethod()> Public Sub setCkWeighted(ByVal SessionGuid$, ByVal vCkWeighted As Boolean) 8](#_Toc246917328)

[<WebMethod()> Public Sub setCkBusiness(ByVal SessionGuid$, ByVal vCkBusiness As Boolean) 8](#_Toc246917329)

[<WebMethod()> Public Sub setTxtSearch(ByVal SessionGuid$, ByVal vTxtSearch As String) 8](#_Toc246917330)

[<WebMethod()> Public Sub setCkLimitToExisting(ByVal SessionGuid$, ByVal vCkLimitToExisting As Boolean) 8](#_Toc246917331)

[<WebMethod()> Public Sub setCkIncludeAttachments(ByVal SessionGuid$, ByVal vckIncludeAttachments As Boolean) 8](#_Toc246917332)

[<WebMethod()> Public Sub setCkDays(ByVal SessionGuid$, ByVal vCkDays As Boolean) 8](#_Toc246917333)

[<WebMethod()> Public Sub setTxtDays(ByVal SessionGuid$, ByVal vTxtDays As String) 8](#_Toc246917334)

[<WebMethod()> Public Sub setCbFromAddr(ByVal SessionGuid$, ByVal vCbFromAddr As String) 8](#_Toc246917335)

[<WebMethod()> Public Sub setCbFromName(ByVal SessionGuid$, ByVal vCbFromName As String) 8](#_Toc246917336)

[<WebMethod()> Public Sub setCbToAddr(ByVal SessionGuid$, ByVal vCbToAddr As String) 8](#_Toc246917337)

[<WebMethod()> Public Sub setCbToName(ByVal SessionGuid$, ByVal vCbToName As String) 8](#_Toc246917338)

[<WebMethod()> Public Sub setRbMyEmails(ByVal SessionGuid$, ByVal vRbMyEmails As Boolean) 8](#_Toc246917339)

[<WebMethod()> Public Sub setCbFolderFilter(ByVal SessionGuid$, ByVal vCbFolderFilter As String) 8](#_Toc246917340)

[<WebMethod()> Public Sub setCbCCaddr(ByVal SessionGuid$, ByVal vCbCCaddr As String) 8](#_Toc246917341)

[<WebMethod()> Public Sub setTxtCCPhrase(ByVal SessionGuid$, ByVal vTxtCCPhrase As String) 8](#_Toc246917342)

[<WebMethod()> Public Sub setbDateSelection(ByVal SessionGuid$, ByVal vCbDateSelection As String) 8](#_Toc246917343)

[<WebMethod()> Public Sub setCkSentDate(ByVal SessionGuid$, ByVal vCkSentDate As Boolean) 8](#_Toc246917344)

[<WebMethod()> Public Sub setDateMailDateEnd(ByVal SessionGuid$, ByVal vDtMailDateEnd As String) 8](#_Toc246917345)

[<WebMethod()> Public Sub setCbDateSelection(ByVal SessionGuid$, ByVal vCbDateSelection As String) 8](#_Toc246917346)

[<WebMethod()> Public Sub setCkCreationDate(ByVal SessionGuid$, ByVal vCkCreationDate As Boolean) 8](#_Toc246917347)

[<WebMethod()> Public Sub setDateMailDateStart(ByVal SessionGuid$, ByVal vDtMailDateStart As String) 8](#_Toc246917348)

[<WebMethod()> Public Sub setCkReceivedTime(ByVal SessionGuid$, ByVal vCkReceivedTime As Boolean) 8](#_Toc246917349)

[<WebMethod()> Public Sub setLibraryName(ByVal SessionGuid$, ByVal vLibraryName As String) 8](#_Toc246917350)

[<WebMethod()> Public Sub setBIncludeLibraryFilesInSearch(ByVal SessionGuid$, ByVal vBIncludeLibraryFilesInSearch As Boolean) 8](#_Toc246917351)

[<WebMethod()> Public Sub setCkSearchMasterDocsOnly(ByVal SessionGuid$, ByVal vCkSearchMasterDocsOnly As Boolean) 8](#_Toc246917352)

[<WebMethod()> Public Sub setCbFilename(ByVal SessionGuid$, ByVal vCbFilename As String) 8](#_Toc246917353)

[<WebMethod()> Public Sub setCbFileTypes(ByVal SessionGuid$, ByVal vCbFileTypes As String) 8](#_Toc246917354)

[<WebMethod()> Public Sub setCbDirs(ByVal SessionGuid$, ByVal vCbDirs As String) 8](#_Toc246917355)

[<WebMethod()> Public Sub setMetaDataSearchKey1(ByVal SessionGuid$, ByVal vCbMeta1 As String) 8](#_Toc246917356)

[<WebMethod()> Public Sub setMetaDataSearchValue1(ByVal SessionGuid$, ByVal vTxtMetaSearch1 As String) 8](#_Toc246917357)

[<WebMethod()> Public Sub setMetaDataSearchKey2(ByVal SessionGuid$, ByVal vCbMeta2 As String) 8](#_Toc246917358)

[<WebMethod()> Public Sub setMetaDataSearchValue2(ByVal SessionGuid$, ByVal vTxtMetaSearch2 As String) 8](#_Toc246917359)

[<WebMethod()> Public Sub setMetaDataSearchKey3(ByVal SessionGuid$, ByVal vCbMeta3 As String) 8](#_Toc246917360)

[<WebMethod()> Public Sub setMetaDataSearchValue3(ByVal SessionGuid$, ByVal vTxtMetaSearch3 As String) 8](#_Toc246917361)

[<WebMethod()> Public Sub setMetaDataSearchKey4(ByVal SessionGuid$, ByVal vCbMeta4 As String) 8](#_Toc246917362)

[<WebMethod()> Public Sub setMetaDataSearchValue4(ByVal SessionGuid$, ByVal vTxtMetaSearch4 As String) 8](#_Toc246917363)

[<WebMethod()> Public Sub setDateCreateDateStart(ByVal SessionGuid$, ByVal vDtCreateDateStart As String) 8](#_Toc246917364)

[<WebMethod()> Public Sub setDateCreateDateEnd(ByVal SessionGuid$, ByVal vDtCreateDateEnd As String) 8](#_Toc246917365)

[<WebMethod()> Public Sub setDateLastWriteEnd(ByVal SessionGuid$, ByVal vDtLastWriteEnd As String) 8](#_Toc246917366)

[<WebMethod()> Public Sub setDateLastWriteStart(ByVal SessionGuid$, ByVal vDtLastWriteStart As String) 8](#_Toc246917367)

[<WebMethod()> Public Sub setCbEvalCreateTime(ByVal SessionGuid$, ByVal vCbEvalCreateTime As String) 8](#_Toc246917368)

[<WebMethod()> Public Sub setCbEvalWriteTime(ByVal SessionGuid$, ByVal vCbEvalWriteTime As String) 8](#_Toc246917369)

[<WebMethod()> Public Sub setGurrUserLoginID(ByVal SessionGuid$, ByVal vGurrUserLoginID As String) 8](#_Toc246917370)

[<WebMethod()> Sub WriteErr(ByVal StmtID As Integer, ByVal PgmName$, ByVal UserID$, ByVal ErrMsg As String) 8](#_Toc246917371)

[<WebMethod()> Function CredentialInfo() As String 8](#_Toc246917372)

[<WebMethod()> Public Function isLicenseValid(ByVal SessionGuid$) As Boolean 8](#_Toc246917373)

[<WebMethod()> Public Function getLicenseStatus(ByVal SessionGuid$) As String 8](#_Toc246917374)

[<WebMethod()> Public Function getUserName() As String 8](#_Toc246917375)

[<WebMethod()> Public Function getDomainName() As String 8](#_Toc246917376)

[<WebMethod()> Public Function getManagerName() As String 8](#_Toc246917377)

[<WebMethod()> Public Function getCurrentThesaurus(ByVal SessionGuid$) As String 8](#_Toc246917378)

[<WebMethod()> Public Sub setCurrentThesaurus(ByVal SessionGuid$, ByVal ThesaurusName As String) 8](#_Toc246917379)

# The purpose of the ECM API

There are times when it may be advantageous to run searches apart from the ECM Library Client. The ECM Client is designed to provide maximum power and full search flexibility. The API, frankly, is not. The API is designed to provide a “basic” interface into the search engine of ECM Library. If you want to exercise the full power of ECM, you will have to use the ECM Light Client from the workstation.

However, the API will certainly provide basic search capabilities and allow interface from any program that can call a web service.

It is our desire to provide our clients with their requested flexibility, the ability to interface to the search engine through an API. It is our recommendation that the ECM Library Client be used instead as it provides maximum utilization of ECM Library’s full power.

# Deploying the web services

***Install ECM Library:***

[ECM Library](http://www.ecmlibrary.com/Download/Ecm2010/publish.htm)

***Install the ECM License***

1. From the License Server using the License Wizard
2. Or from the EMAIL, cut and paste the license text

***Install ECM Web Services***

[Download and Install Web Services](http://www.ecmlibrary.com/download/EcmWebServicesSetup/setup.exe)

# What is lost by searching thru a service and not through the ECM client?

1. Cannot search on the Thesaurus, can only search on the “Microsoft Thesaurus”.
2. Without forcing login to ECM Library, the ability to determine whether a user is an admin, super admin, power searcher or default user is lost.
3. Without forcing login to ECM Library, the ability for a user to search only on his/her content is lost.
4. Without forcing login to ECM Library, any content not marked “public” cannot be searched.

# ECM Web Services Calls:

This SDK is designed to work with both client based, Windows Form applciations as well as, Browser Based applciations. If your calling program can access a web service, then ECM Web Services are availble to that calling routine.

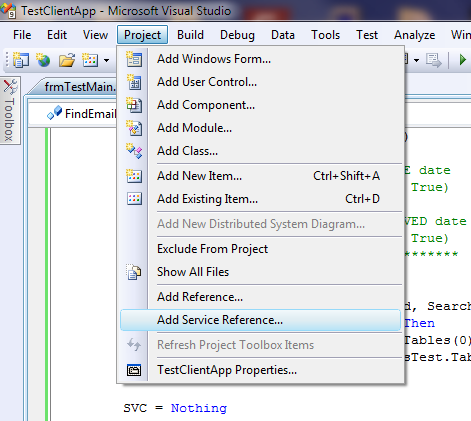
Because Session Variables are not availble to client based applciations and are almost required by browser based applications, ECM Web Services has provided this capability in a unique fashion. ECM uses a user supplied GUID as a session ID which allows effective emmulation of a Session ID in both client and web applciations. All of our example code is supplied in VB.NET syntax.

Therefore, whenever it is needed to track a single execution instance, a Session ID, in our case a session GUID, should be provided. To do so:

Dim SessionGuid$ = Guid.NewGuid.ToString

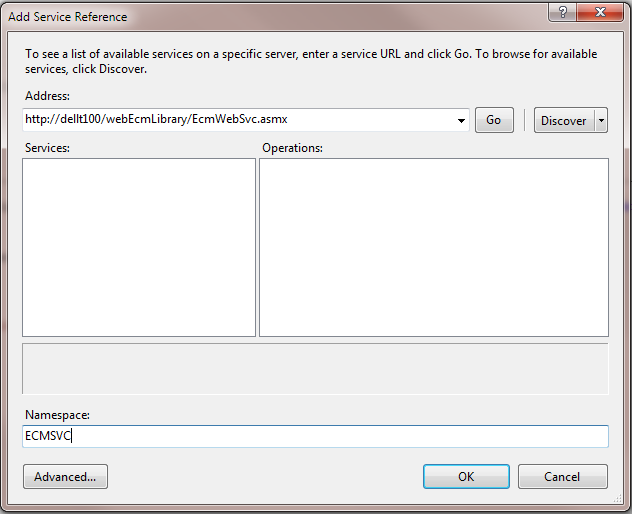
## Installing a reference to the service

In order to call ECM Web services, they must be installed on a server running IIS 7.0 or later. Once installed, add a refernce to the services within your development environment.



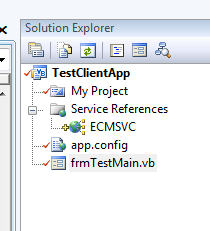
Using Visual Studio, there are a couple of ways to do this. In the method above, we went to the Project tab, then to the Add Service Reference menu item. Click on the menu item.

This will open up the Serivces definition screen:



Enter the address of your server and the reference to ECM Web Services. In the bottom line, enter the name you wish to call the reference. In our case, we entered ECMSVC. Press the OK button and if the server is configured properly with the Web Services and you have the correct authority, a connection should be established.

You should then see in under the Solution View,



This indicates that you now have full access to the ECM Web Services.

# The Details of the Web Services

***This function is not exposed.***

**getLicenseStatus**

Parameters:

SessionGuid Same as a session variable.

Prototype:

Function getLicenseStatus(byVal SessionGuid$) As String

**Purpose:**

This is an internal function that is called to validate the status of current ECM license. ECM library will continue to run wihout validation for the life of the existing license if it is leased or indefinitely if it is a purchased license. At the end of the “lease” time, without a connection to the ECM Server, the license will expire and ECM will shut down. A license can easily be re-instated, upgraded, or modified with a call to ECM Support. A new or modified license can be delived by FTP, Email, or connecting to the ECM License Server.

If it is a purchased license, the service will continue to run uninterrupted. If the number of purchased seats is exceeded, the system will only allow ***registered machines*** to login to the system. In the case of a roaming license, once the number of users is exceeded, only existing users can log in from already defined machines.

Example Code:

NONE Provided – unexposed function. For internal use only.

# Exposed Web Services

**WriteErr**

Prototype:

Public sub WriteErr(ByVal StmtID As Integer, ByVal PgmName$, ByVal UserID$, ByVal ErrMsg As String)

**Purpose:**

Call this routine to write an error message to the ECM Services Log. This log resides within the repository and should be managed as all other highly active tables.

**Example Code:**

'\*\* No session id is needed here

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim StatementId As String = "1001.a"

Dim ProgramName As String = "TestPgm"

Dim UserIdentifier As String = System.Net.Dns.GetHostName()

SVC.WriteErr(StatementId, ProgramName, UserIdentifier, "This is the error message to log.")

SVC = Nothing

**restoreDocument**

Parameters:

SessionGuid$ Session GUID is used as a SESSION ID across the web.

SourceGuid$

SavePath$

OverWrite

Prototype:

Public Function restoreDocument(ByVal SessionGuid$, ByVal SourceGuid$, ByVal SavePath$, ByVal OverWrite As Boolean) As Boolean

**Purpose:**

In order to restore source, the repository GUID of that content is required. Pass the GUID and the path (the directory) to where you wish the restored content to be placed. If the content already exists, and you wish to overwrite it, just set the Over Write to true.

Example Code:

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim RestoreDirectory As String = "c:\temp"

Dim OverwriteDocumentIfExtsts As Boolean = True

Dim UserIdentifier As String = System.Net.Dns.GetHostName()

SVC.restoreDocument(SessionGuid$, RestoreDirectory, OverwriteDocumentIfExtsts)

SVC = Nothing

**restoreEmail**

Parameters:

SourceGuid $

EmailGuid $

SavePath$

OverWrite

Prototype:

Function restoreEmail(ByVal SourceGuid$, ByVal EmailGuid$, ByVal SavePath$, ByVal OverWrite As Boolean) As Boolean

**Purpose:**

In order to restore EMAIL, the repository GUID of that EMAIL is required. Pass the GUID and the path (the directory) to where you wish the restored content to be placed. If the content already exists, and you wish to overwrite it, just set the Over Write to true.

Example Code:

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim RestoreDirectory As String = "c:\temp"

Dim OverwriteDocumentIfExtsts As Boolean = True

Dim UserIdentifier As String = System.Net.Dns.GetHostName()

SVC. restoreEmail (SourceGuid $, RestoreDirectory, OverwriteDocumentIfExtsts)

SVC = Nothing

**getConnectionString**

Parameters:

No parameters required.

Prototype:

Function getConnectionString() As String

**Purpose:**

In order to communicate with the ECM Repository it is necessary to use SQL connection strings. This allows maximum flexibility in pulling from and searching any and all ECM repositories. Call this function to get the current connection string.

Example Code:

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim ConnectionStr As String = SVC.getConnectionString()

MsgBox(ConnectionStr)

SVC = Nothing

**genDocSearchSql**

Parameters:

SessionGuid$ Session GUID is used as a SESSION ID across the web.

SearchString$ The search string using ECM Library syntax.

Prototype:

Public Function genDocSearchSql(ByVal SessionGuid$, ByVal SearchString$) As String

**Purpose:**

At times, it is helpful to see the search SQL genereated by ECM Library. Call this funciton and it will return a string variable containing the generated SQL for a content search. It will not display the underlying code that accesses the ECM Neural Index.

Example Code:

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim SearchCriteria As String = "alaska +hunting +firearms"

Dim OverwriteDocumentIfExtsts As Boolean = True

Dim UserIdentifier As String = System.Net.Dns.GetHostName()

Dim GeneratedSql As String = SVC.genDocSearchSql(SessionGuid, SearchCriteria)

MsgBox(GeneratedSql)

SVC = Nothing

**genEmailSearchSql**

Parameters:

SessionGuid$ Session GUID is used as a SESSION ID across the web.

SearchString$ The search string using ECM Library syntax.

Prototype:

Public Function genEmailSearchSql(ByVal SessionGuid$, ByVal SearchString$) As String

**Purpose:**

At times, it is helpful to see the search SQL genereated by ECM Library. Call this funciton and it will return a string variable containing the generated SQL for a email search.

Example Code:

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim SearchCriteria As String = "alaska +hunting +firearms"

Dim OverwriteDocumentIfExtsts As Boolean = True

Dim UserIdentifier As String = System.Net.Dns.GetHostName()

Dim GeneratedSql As String = SVC.genEmailSearchSql (SessionGuid, SearchCriteria)

MsgBox(GeneratedSql)

SVC = Nothing

**SearchEmail**

Parameters:

SessionGuid$ Session GUID is used as a SESSION ID across the web.

SearchString$ The search string using ECM Library syntax.

Prototype:

Public Function SearchEmail(ByVal SessionGuid$, ByVal SearchString$) As DataSet

**Purpose:**

Example Code:

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim SearchText As String = txtSearch.Text

If SearchText.Trim.Length = 0 Then

MsgBox("Please fill in the search line, returning.")

Return

End If

Dim dsTest As New DataSet

dsTest = SVC.SearchEmail(SessionGuid, SearchText)

If dsTest.Tables(0).Rows.Count > 0 Then

SB.Text = "Returned#:" + dsTest.Tables(0).Rows.Count.ToString

Me.DataGridView1.DataSource = dsTest.Tables(0)

End If

SVC = Nothing

**SearchEmailXML**

Parameters:

SessionGuid$ Session GUID is used as a SESSION ID across the web.

SearchString$ The search string using ECM Library syntax.

Prototype:

Public Function SearchEmailXML(ByVal SessionGuid$, ByVal SearchString$) As String

**Purpose:**

Example Code:

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim SearchText As String = txtSearch.Text

Dim strXML As String = ""

If SearchText.Trim.Length = 0 Then

MsgBox("Please fill in the search line, returning.")

Return

End If

Dim dsTest As New DataSet

strXML = SVC.SearchEmailXML(SessionGuid, SearchText)

MsgBox(strXML)

SVC = Nothing

**setConnStr**

***Note: There should never really be a need to use this fucntion when using a single ECM repository. It is provided for those that run multiple repositories on multiple servers and need to programatically switch between them.***

Parameters:

SessionGuid

ConnStr

Prototype:

Public Function setConnStr(byVal SessionGuid$, ByVal ConnStr$) As Boolean

**Purpose:**

In order to communicate with the ECM Repository it is necessary to use SQL connection strings. This allows maximum flexibility in pulling from and searching any and all ECM repositories. Call this function to SET the current connection string and check the return code is TRUE for success or FALSE for failure.

Example Code:

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim RestoreDirectory As String = "c:\temp"

Dim OverwriteDocumentIfExtsts As Boolean = True

Dim UserIdentifier As String = System.Net.Dns.GetHostName()

Dim NewConnectionStr$ = "Provider=SQLOLEDB;Data Source= SERVER\_NAME;Integrated Security=SSPI;Initial Catalog=ECM.Library"

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim B As Boolean = SVC.setConnStr(SessionGuid, NewConnectionStr$)

If B Then

MsgBox("New Connection String successfully set.")

End If

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim SearchText As String = txtSearch.Text

Dim strXML As String = ""

If SearchText.Trim.Length = 0 Then

MsgBox("Please fill in the search line, returning.")

Return

End If

'\*\* Search using the new connection string

Dim dsTest As New DataSet

strXML = SVC.SearchEmailXML(SessionGuid, SearchText)

MsgBox(strXML)

SVC = Nothing

**getConnectionString**

Prototype:

Public Function getConnStr(byVal SessionGuid$, ) As String

**Purpose:**

This is the short hand for of getConnectionString.

Example Code:

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim ConnectionString = SVC.getConnectionString()

MsgBox("Connection String: " + ConnectionString)

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SVC = Nothing

**getSqlDS**

Prototype:

Public Function getSqlDS(byVal SessionGuid$, ByVal SelectSql As String, ByVal ConnStr$) As dataset

**Purpose:**

This function takes any “well-form sql” and will produce a return dataset. This dataset can be used by your application.

Example Code:

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim MySql As String = "Select UserName, UserLoginID from Users order by UserName"

Dim NewConnectionStr$ = "Provider=SQLOLEDB;Data Source=DELLT100;Integrated Security=SSPI;Initial Catalog=ECM.Library"

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim B As Boolean = SVC.setConnStr(SessionGuid, NewConnectionStr$)

If B Then

MsgBox("Connection String successfully set.")

End If

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim SearchText As String = txtSearch.Text

Dim strXML As String = ""

If SearchText.Trim.Length = 0 Then

MsgBox("Please fill in the search line, returning.")

Return

End If

'\*\* Search using the new connection string

Dim dsTest As New DataSet

dsTest = SVC.getSqlDS(MySql, NewConnectionStr)

MsgBox("Returned " + dsTest.Tables(0).Rows.Count.ToString + " rows.")

SVC = Nothing

**getSqlDsXML**

Prototype:

Public Function getSqlDsXML(ByVal SelectSql As String, ByVal ConnStr$) As String

**Purpose:**

This function takes any “well-form sql” and will produce a return and XML dataset. This XML dataset can be used by your application.

Example Code:

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim MySql As String = "Select UserName, UserLoginID from Users order by UserName"

Dim NewConnectionStr$ = "Provider=SQLOLEDB;Data Source=DELLT100;Integrated Security=SSPI;Initial Catalog=ECM.Library"

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim B As Boolean = SVC.setConnStr(SessionGuid, NewConnectionStr$)

If B Then

MsgBox("Connection String successfully set.")

End If

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\* Search using the new connection string

Dim strXml$ = ""

strXml$ = SVC.getSqlDsXML(MySql, NewConnectionStr)

MsgBox(strXml)

SVC = Nothing

**iCountRecordsInTable**

Parameters:

Tbl The table name in the EC.Library database.

Prototype:

Public Function iCountRecordsInTable(ByVal Tbl As String) As Integer

**Purpose:**

This function takes the name of any ECM table and will return only the count of records contained within that table.

Example Code:

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim iRows As Integer = SVC.iCountRecordsInTable("DataSource")

MsgBox("#Rows in table DataSource " = iRows.ToString)

SVC = Nothing

**getSourceImage**

Parameters:

SourceGuid

SourceLength

Prototype:

Public Function getSourceImage(ByVal SourceGuid$, ByRef SourceLength As Integer) As Byte()

**Purpose:**

This function is called to retrieve source content from the repository. Content within ECM Library is stored as image files. Calling this function and passing in the GUID of the desired content will return a binary array of data that can be used by your application.

Example Code:

**getEmailImage**

Parameters:

XX

Prototype:

Public Function getEmailImage(ByVal EmailGuid$, ByRef SourceLength As Integer) As Byte()

**Purpose:**

This function is called to retrieve emails from the repository. EMAILS within ECM Library are stored as image files. Calling this function and passing in the GUID of the desired email will return a binary array of data that can be used by your application.

Example Code:

**GetMachineIPAddr**

Parameters:

XX

Prototype:

Public Function GetMachineIPAddr(byVal SessionGuid$, ) As String

**Purpose:**

This function returns the IP address of the invoking machine. Such that, if you access ECM across the web it will pass back the IP of the calling machine.

**setSearchString**

Parameters:

XX

Prototype:

Public Sub setSearchString(byVal SessionGuid$, ByVal SearchString As String)

**Purpose:**

This is used to set an Email and Content Search parameter.

This functions passes the users search string into ECM Library for processing.

**getSearchString**

Parameters:

XX

Prototype:

Public Function getSearchString(byVal SessionGuid$, ) As String

**Purpose:**

This is used to set an Email and Content Search parameter.

This functions retrieves the users current search string from ECM Library and returns it a a string.

**setIncludeAttachments**

Parameters:

XX

Prototype:

Public Sub setIncludeAttachments(byVal SessionGuid$, ByVal IncludeAttachments As Boolean)

**Purpose:**

This is used to set an Email Search parameter.

Set this variable to TRUE to include email attachments in your search.

**setIsAdmin**

Parameters:

XX

Prototype:

Public Sub setIsAdmin(byVal SessionGuid$, ByVal vIsAdmin As Boolean)

**Purpose:**

Set this variable to TRUE to set the current USER to have admin authority within ECM.

**getIsAdmin**

Parameters:

XX

Prototype:

public Function getIsAdmin(byVal SessionGuid$, ) As Boolean

**Purpose:**

Returns the whether the current user is an administrator or not; True if so, false if not.

**setIsGlobalSearcher**

Parameters:

XX

Prototype:

Public Sub setIsGlobalSearcher(byVal SessionGuid$, ByVal vIsGlobalSearcher As Boolean)

**Purpose:**

Set this variable to TRUE to set the current USER to have Global Searcher authority within ECM.

**setCurrUserGuidID**

Parameters:

XX

Prototype:

Public Sub setCurrUserGuidID(byVal SessionGuid$, ByVal vCurrUserGuidID As String)

**Purpose:**

All users defined to ECM have their user ID and an associated GUID. Call this function to set the current user’s GUID. This allows secure navigation and selection of content.

**setMasterContentOnly**

Parameters:

SessionGuid

vMasterContentOnly This is a Boolean indicator that is set to either true or false

Prototype:

Public Sub setMasterContentOnly(byVal SessionGuid$, ByVal vMasterContentOnly As Boolean)

**Purpose:**

This is used to set an Email and Content Search parameter.

Set this variable to TRUE to search only documents marked as Master Documents.

Example Code:

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim SearchText As String = txtSearch.Text

If SearchText.Trim.Length = 0 Then

MsgBox("Please fill in the search line, returning.")

Return

End If

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\* This will allow the seach to only return MASTER

‘\*\* content.

VC. setMasterContentOnly (SessionGuid, True)

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim dsTest As New DataSet

dsTest = SVC.SearchDocuments(SessionGuid$, SearchText)

If dsTest.Tables(0).Rows.Count > 0 Then

SB.Text = "Returned " + dsTest.Tables(0).Rows.Count.ToString + " rows."

Me.DataGridView1.DataSource = dsTest.Tables(0)

End If

SVC = Nothing

**setMyContentOnly**

Parameters:

SessionGuid

vMyContentOnly This is a Boolean indicator that is set to either true or false

Prototype:

Public Sub setMyContentOnly(byVal SessionGuid$, ByVal vMyContentOnly As Boolean)

**Purpose:**

This is used to set an Email and Content Search parameter.

Set this variable to TRUE to search only content owned by the current user.

Example Code:

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim SearchText As String = txtSearch.Text

If SearchText.Trim.Length = 0 Then

MsgBox("Please fill in the search line, returning.")

Return

End If

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\* This will allow the seach to only return content

'\*\* belonging to the currently logged in user/machine.

SVC.setMyContentOnly(SessionGuid, True)

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim dsTest As New DataSet

dsTest = SVC.SearchDocuments(SessionGuid$, SearchText)

If dsTest.Tables(0).Rows.Count > 0 Then

SB.Text = "Returned " + dsTest.Tables(0).Rows.Count.ToString + " rows."

Me.DataGridView1.DataSource = dsTest.Tables(0)

End If

SVC = Nothing

**setsGlobalSearcher**

Parameters:

XX

Prototype:

Public Sub setsGlobalSearcher(byVal SessionGuid$, ByVal vIsGlobalSearcher As Boolean)

**Purpose:**

Set this variable to TRUE to set the current USER to have Global Searcher authority within ECM.

**setNbrOfErrors**

Parameters:

XX

Prototype:

Public Sub setbrOfErrors(byVal SessionGuid$, ByVal vNbrOfErrors As Integer)

**Purpose:**

Set this variable to increment as errors are encountered for any reason. It is a global variable that tracks the number of errors encountered within a session.

**setSql**

Parameters:

XX

Prototype:

Public Sub setSql(byVal SessionGuid$, ByVal vSql As String)

**Purpose:**

Allows a SQL statement to be instantiated. It can then be executed later thru anohter service.

**setErrMsg**

Parameters:

XX

Prototype:

Public Sub setErrMsg(byVal SessionGuid$, ByVal vErrMsg As String)

**Purpose:**

Set an error mesasge to be recorded or presented.

**setMinRating**

Parameters:

XX

Prototype:

Public Sub setMinRating(byVal SessionGuid$, ByVal vMinRating As Integer)

**Purpose:**

This is used to set an Email and Content Search parameter.

Set this number to be the MINIMUM weight required before it will be presented or returned from a query.

**setClipBoardData**

Parameters:

XX

Prototype:

Public Sub setClipBoardData(byVal SessionGuid$, ByVal vClipBoardData As String)

**Purpose:**

The clipboard function is not available in web apps. However, this emmulates that fucntion in that you can set clipboard text and reacall it later.

**setCkLimitToLib**

Parameters:

XX

Prototype:

Public Sub setCkLimitToLib(byVal SessionGuid$, ByVal vCkLimitToLib As Boolean)

**Purpose:**

This is used to set an Email and Content Search parameter.

This is set to true to limit a search to a particualr library.

**setCbLibrary**

Parameters:

XX

Prototype:

Public Sub setCbLibrary(byVal SessionGuid$, ByVal vCbLibrary As String)

**Purpose:**

This is used to set an Email and Content Search parameter.

This is set to to a particualr library’s name and that will be the only library saerched with the prior Ck Limit is set to true.

**setTxtSubject**

Parameters:

XX

Prototype:

Public Sub setTxtSubject(byVal SessionGuid$, ByVal vTxtSubject As String)

**Purpose:**

This is used to set an Email Search parameter.

When it is desirable to searh for an email’s subject, call this function. Wild cards can be used such as ‘%ECM%’ would effect a search against all email subjects that contained the letters ECM any where within the text.

**setCkWeighted**

Parameters:

XX

Prototype:

Public Sub setCkWeighted(byVal SessionGuid$, ByVal vCkWeighted As Boolean)

**Purpose:**

This is used to set an Email and Content Search parameter.

Set to true to use weights in the search.

**setCkBusiness**

Parameters:

XX

Prototype:

Public Sub setCkBusiness(byVal SessionGuid$, ByVal vCkBusiness As Boolean)

**Purpose:**

This is used to set an Email and Content Search parameter.

Set to true to use a very free form type of search known as FREETEXT. We do not advise this as the resulting query can take an exceptionally long time to execute. But is can be used when not much is known about the document for which you are searching. It is slow and expansive.

**setTxtSearch**

Parameters:

XX

Prototype:

Public Sub setTxtSearch(byVal SessionGuid$, ByVal vTxtSearch As String)

**Purpose:**

This is used to set an Email and Content Search parameter.

Pass in the search parameters.

**setCkLimitToExisting**

Parameters:

XX

Prototype:

Public Sub setCkLimitToExisting(byVal SessionGuid$, ByVal vCkLimitToExisting As Boolean)

**Purpose:**

This is used to set an Email and Content Search parameter.

When set true, the search becomes a reduction search and will only include content already existing in the previous search.

**setckIncludeAttachments**

Parameters:

XX

Prototype:

Public Sub setckIncludeAttachments(byVal SessionGuid$, ByVal vckIncludeAttachments As Boolean)

**Purpose:**

This is used to set an Email Search parameter.

Set to true to include email attachments in the search.

**setCkDays**

Parameters:

XX

Prototype:

Public Sub setCkDays(byVal SessionGuid$, ByVal vCkDays As Boolean)

**Purpose:**

This is used to set an Email Search parameter.

Set to true to limit the retued search to a specific number of days prior to today.

**setTxtDays**

Parameters:

XX

Prototype:

Public Sub setTxtDays(byVal SessionGuid$, ByVal vTxtDays As String)

**Purpose:**

This is used to set an Email Search parameter.

Pass in the number of days desired to search back in time. For example if 90 waS PASSED in, the search would only consider content created/entered within the past 90 days.

**setCbFromAddr**

Parameters:

XX

Prototype:

Public Sub setCbFromAddr(byVal SessionGuid$, ByVal vCbFromAddr As String)

**Purpose:**

This is used to set an Email Search parameter.

If all or part of a “from” email address is to be searched, set the value here. Wild cards are acceptable.

**setCbFromName**

Parameters:

XX

Prototype:

Public Sub setCbFromName(byVal SessionGuid$, ByVal vCbFromName As String)

**Purpose:**

This is used to set an Email Search parameter.

If all or part of a “from name” is to be searched, set the value here. Wild cards are acceptable.

**setCbToAddr**

Parameters:

XX

Prototype:

Public Sub setCbToAddr(byVal SessionGuid$, ByVal vCbToAddr As String)

**Purpose:**

This is used to set an Email Search parameter.

If all or part of a “to email address” is to be searched, set the value here. Wild cards are acceptable. This is a reduction search.

**setCbToName**

Parameters:

XX

Prototype:

Public Sub setCbToName(byVal SessionGuid$, ByVal vCbToName As String)

**Purpose:**

This is used to set an Email Search parameter.

If all or part of a “to name” is to be searched, set the value here. Wild cards are acceptable. This is a reduction search.

**setRbMyEmails**

Parameters:

XX

Prototype:

Public Sub setRbMyEmails(byVal SessionGuid$, ByVal vRbMyEmails As Boolean)

**Purpose:**

This is used to set an Email Search parameter.

Set to true to search only your emails.

**setCbFolderFilter**

Parameters:

XX

Prototype:

Public Sub setCbFolderFilter(byVal SessionGuid$, ByVal vCbFolderFilter As String)

**Purpose:**

This is used to set an Email Search parameter.

If all or part of an “email folder name” is to be searched, set the value here. Wild cards are acceptable. This is a reduction search.

**setCbCCaddr**

Parameters:

XX

Prototype:

Public Sub setCbCCaddr(byVal SessionGuid$, ByVal vCbCCaddr As String)

**Purpose:**

This is used to set an Email Search parameter.

If all or part of an “CC email address” is to be searched, set the value here. Wild cards are acceptable. This is a reduction search.

**setTxtCCPhrase**

Parameters:

XX

Prototype:

Public Sub setTxtCCPhrase(byVal SessionGuid$, ByVal vTxtCCPhrase As String)

**Purpose:**

This is used to set an Email Search parameter.

If all or part of an “CC email address” is to be searched, set the value here. Wild cards are acceptable. This is a reduction search.

**setCbDateSelection**

Parameters:

SessionGuid:

vCbDateSelection - One of four values OFF,After,Before,Between,Not Between.

Prototype:

Public Sub setCbDateSelection(byVal SessionGuid$, ByVal vCbDateSelection As String)

**Purpose:**

Set the date limiting criteria for an EMAIL search.

There are times when it is desirable to search before, after or bbetween dates. The entries here are NULL (off), Before, After, or between.

**setCkSentDate**

Parameters:

XX

Prototype:

Public Sub setCkSentDate(byVal SessionGuid$, ByVal vCkSentDate As Boolean)

**Purpose:**

This is used to set an Email Search parameter.

There are times when it is desirable to search before, after or between dates. This is where the “Start Range” Mail Sent Date is set for searching.

This is used to set an Email Search parameter.

**setDateMailDateEnd**

Parameters:

XX

Prototype:

Public Sub setDateMailDateEnd(byVal SessionGuid$, ByVal vDtMailDateEnd As String)

**Purpose:**

This is used to set an Email Search parameter.

There are times when it is desirable to search before, after or between dates. This is where the “END Range” Mail Sent Date is set for searching.

**setCbDateSelection**

Allowed Values:

OFF

After

Before

Between

Not Between

On

Parameters:

XX

Prototype:

Public Sub setCbDateSelection(byVal SessionGuid$, ByVal vCbDateSelection As String)

**Purpose:**

This is used to set an Email Search parameter.

This value is used to:

**setCkCreationDate**

Parameters:

XX

Prototype:

Public Sub setCkCreationDate(byVal SessionGuid$, ByVal vCkCreationDate As Boolean)

**Purpose:**

This is used to set an Email Search parameter.

There are times when it is desirable to search before, after or between dates. This is where the “Start Range” Mail Created Date is set for searching.

**setDateMailDateStart**

Parameters:

XX

Prototype:

Public Sub setDateMailDateStart(byVal SessionGuid$, ByVal vDtMailDateStart As String)

**Purpose:**

This is used to set an Email Search parameter.

There are times when it is desirable to search before, after or between dates. This is where the “Start Range” Mail Sent Date is set for searching.

**setCkReceivedTime**

Parameters:

XX

Prototype:

Public Sub setCkReceivedTime(byVal SessionGuid$, ByVal vCkReceivedTime As Boolean)

**Purpose:**

This is used to set an Email Search parameter.

Set to true to

**setLibraryName**

Parameters:

XX

Prototype:

Public Sub setLibraryName(byVal SessionGuid$, ByVal vLibraryName As String)

**Purpose:**

This is used to set an Email and a Content Search parameter.

When selected to search a specific library only, this is where the name of that library is supplied.

**setBIncludeLibraryFilesInSearch**

Parameters:

XX

Prototype:

Public Sub setBIncludeLibraryFilesInSearch(byVal SessionGuid$, ByVal vBIncludeLibraryFilesInSearch As Boolean)

**Purpose:**

This is used to set an Email Search and a Content parameter.

When selected to search a specific source directory, this is where the name of that directory is supplied. Wild cards are acceptable.

**setCkSearchMasterDocsOnly**

Parameters:

XX

Prototype:

Public Sub setCkSearchMasterDocsOnly(byVal SessionGuid$, ByVal vCkSearchMasterDocsOnly As Boolean)

**Purpose:**

Set to true to limit CONTENT search to master documents only.

**setCbFilename**

Parameters:

XX

Prototype:

Public Sub setCbFilename(byVal SessionGuid$, ByVal vCbFilename As String)

**Purpose:**

When selected to search a specific file name, this is where the name of that file is supplied. Wild cards are acceptable.

**setCbFileTypes**

Parameters:

XX

Prototype:

Public Sub setCbFileTypes(byVal SessionGuid$, ByVal vCbFileTypes As String)

**Purpose:**

When desired to search for a specific file type, this is where the type is supplied. Wild cards are acceptable.

**setCbDirs**

Parameters:

XX

Prototype:

Public Sub setCbDirs(byVal SessionGuid$, ByVal vCbDirs As String)

**Purpose:**

**setMetaDataSearchKey**1

Parameters:

XX

Prototype:

Public Sub setMetaDataSearchKey1(byVal SessionGuid$, ByVal vCbMeta1 As String)

**Purpose:**

When selecte to search for a specific metadata name, this is where the name of that metadata is supplied. Wild cards are acceptable.

**setMetaDataSearchValue**1

Parameters:

XX

Prototype:

Public Sub setMetaDataSearchValue1(byVal SessionGuid$, ByVal vTxtMetaSearch1 As String)

**Purpose:**

This is the associated value that will be related to the above metadata. For example, if you wish to limit FILESIZE to < 40000, this is where the 40000 would go.

**setMetaDataSearchKey**2

Parameters:

XX

Prototype:

Public Sub setMetaDataSearchKey2(byVal SessionGuid$, ByVal vCbMeta2 As String)

**Purpose:**

When selected to search for a specific metadata name, this is where the name of that metadata is supplied. Wild cards are acceptable.

**setMetaDataSearchValue**2

Parameters:

XX

Prototype:

Public Sub setMetaDataSearchValue2(byVal SessionGuid$, ByVal vTxtMetaSearch2 As String)

**Purpose:**

This is the associated value that will be related to the above metadata. For example, if you wish to limit FILESIZE to < 40000, this is where the 40000 would go.

**setMetaDataSearchKey**3

Parameters:

XX

Prototype:

Public Sub setMetaDataSearchKey3(byVal SessionGuid$, ByVal vCbMeta3 As String)

**Purpose:**

When selected to search for a specific metadata name, this is where the name of that metadata is supplied. Wild cards are acceptable.

**setMetaDataSearchValue**3

Parameters:

XX

Prototype:

Public Sub setMetaDataSearchValue3(byVal SessionGuid$, ByVal vTxtMetaSearch3 As String)

**Purpose:**

This is the associated value that will be related to the above metadata. For example, if you wish to limit FILESIZE to < 40000, this is where the 40000 would go.

**setMetaDataSearchKey**4

Parameters:

XX

Prototype:

Public Sub setMetaDataSearchKey4(byVal SessionGuid$, ByVal vCbMeta4 As String)

**Purpose:**

**setMetaDataSearchValue**4

Parameters:

XX

Prototype:

Public Sub setMetaDataSearchValue4(byVal SessionGuid$, ByVal vTxtMetaSearch4 As String)

**Purpose:**

This is the associated value that will be related to the above metadata. For example, if you wish to limit FILESIZE to < 40000, this is where the 40000 would go.

**setDateLastAccessStart**

Parameters:

XX

Prototype:

Public Sub setDateLastAccessStart(byVal SessionGuid$, ByVal vDtCreateDateStart As String)

**Purpose:**

There are times when it is desirable to search before, after or between dates. This is where the “Start Range” Last Access Date is set for searching.

**setDateLastAccessEnd**

Parameters:

XX

Prototype:

Public Sub setDateLastAccessEnd(byVal SessionGuid$, ByVal vDtCreateDateEnd As String)

**Purpose:**

There are times when it is desirable to search before, after or bbetween dates. This is where the “End Range” Last Access Date is set for searching.

**setDateLastWriteEnd**

Parameters:

XX

Prototype:

Public Sub setDateLastWriteEnd(byVal SessionGuid$, ByVal vDtLastWriteEnd As String)

**Purpose:**

There are times when it is desirable to search before, after or bbetween dates. This is where the “End Range” Last Write Date is set for searching.

**setDateLastWriteStart**

Parameters:

XX

Prototype:

Public Sub setDateLastWriteStart(byVal SessionGuid$, ByVal vDtLastWriteStart As String)

**Purpose:**

There are times when it is desirable to search before, after or bbetween dates. This is where the “Start Range” Last Write Date is set for searching.

**setCbEvalCreateTime**

Allowed Values:

OFF

After

Before

Between

Not Between

On

Parameters:

XX

Prototype:

Public Sub setCbEvalCreateTime(byVal SessionGuid$, ByVal vCbEvalCreateTime As String)

**Purpose:**

There are times when it is desirable to search before, after or bbetween dates. The entries here are NULL (off), Before, After, or between.

**setCbEvalWriteTime**

Allowed Values:

OFF

After

Before

Between

Not Between

On

Parameters:

XX

Prototype:

Public Sub setCbEvalWriteTime(byVal SessionGuid$, ByVal vCbEvalWriteTime As String)

**Purpose:**

There are times when it is desirable to search before, after or bbetween dates. The entries here are NULL (off), Before, After, or between.

**setCurrUserLoginID**

Parameters:

XX

Prototype:

Public Sub setGurrUserLoginID(byVal SessionGuid$, ByVal vGurrUserLoginID As String)

Use this function to set the current user’s login ID. It is global and will be tracked for the duration of the session.

**Purpose:**

**SearchDocuments**

Parameters:

XX

Prototype:

Public Function SearchDocuments(byVal SessionGuid$, ByVal SearchString$) As DataSet

**Purpose:**

**SearchDocumentsXML**

Parameters:

XX

Prototype:

Public Function SearchDocumentsXML(byVal SessionGuid$, ByVal SearchString$) As String

**Purpose:**

# *ECM Service Execution Parameters:*

### Mandatory Search Parameters

gGurrUserLoginID As String = null – This is a mandatory field. In the case where no user ID is required, a machine ID will be used in its place. This value allows users to login to the system and tracks seats and users defined to the system

gCurrUserGuidID As String = "" – In order for content to be archived, searched and retrieved, a unique identifier or in the absence of a logged in user, a machine ID is assigned to the “owner” of that content. This unique identifier is a GUID that is related to the User Login ID or machine ID. Both are unique within a repository. The CurrUserGuidID is used to track the relationship of users to content, rights, access, and group/library assignments. It is critical to the operation of the system. In the case where no user ID is required, a machine ID will be used in its place.

gConnStr As String = "" – A connection string is required to attach to an ECM Repository. Set this parameter to the desired repository and all searches will be directed specifically here.

gSearchString As String = "" – This is the “search string” that has to be passed into the web sewrvice. It is the same for eiter email or content search and must follow the standard syntax of an “ECM Search String”. It defaults to a null and if it is not filled in and passed into the search engine, a default NULL search will be returned or an error message.

CkLimitToExisting As Boolean = False – This is a critical part of a reduction search. If set to ‘false’ a search will traverse the entire repository for corresponding emails or content. If set to true, the search wil be limtied to include only the data retured from the last search. This allows a searcher or user to quickly reduce data to a small set of data that includes only the search terms for which they are searching.

### '\*\* Operational Data

IsAdmin As Boolean = False – This is an indicator that can be set to true to indicate that the current user is an administrator. This allows the user to search any content contained within the database. Caution should be excerciused as a user with this priviledge can search on all content and retrieve the same.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public Sub setIsAdmin(ByVal SessionGuid$, ByVal IsAdmin As Boolean)

Public Function getIsAdmin(ByVal SessionGuid$) As Boolean

gIsGlobalSearcher As Boolean = False – Set this to true so that the specified user can search on anything. This user has the same priviledges as an administrator, but cannot delete any records.

**Controlled by:**

Public Sub setIsGlobalSearcher(ByVal SessionGuid$, ByVal IsGlobalSearcher As Boolean)

Public Function getIsGlobalSearcher(ByVal SessionGuid$) As Boolean

gThesaurus As String = "" – The default thesaurus is Roget’s. However, different thesauri can be added as needed to the repository. This is the field where the name(or code) of the desired thesaurus will be entered. Then any words searched by thesaurus will be run against the specified thesaurus.

**Controlled by:**

Public Function getCurrentThesaurus(ByVal SessionGuid$) As String

Public Sub setCurrentThesaurus(ByVal SessionGuid$, ByVal ThesaurusName As String)

gThesaurusWords As String = "" – Set the words that are to be expanded and searched by the thesaurus in this parameter.

Note: This function is not exposed and should not be used. Results are not prdictible.

**Controlled by:**

Public Function getThesaurusWords(ByVal SessionGuid$) As String

Public Sub setThesaurusWords(ByVal SessionGuid$, ByVal ThesaurusWords$)

### Search Data

TargetThesarus As String = "" – This is the name of the thesaurus to be used in an expanded search if other than the default Roget.

**Controlled by:**

Public Sub setTgtThesaurus(ByVal SessionGuid$, ByVal Thesaurus$)

Public Function getTgtThesaurus(ByVal SessionGuid$)

ThesarusWords As String = "" - Set the words that are to be expanded and searched by the thesaurus in this parameter.

**Controlled by:**

Public Sub setThesaurusWords(ByVal SessionGuid$, ByVal Thesaurus$)

### Database Class Parameters

gConn As SqlConnection – Internal ECM global database connection. This is not an exposed parameter.

gThesauri As New ArrayList – This is the name of the global thesaurus to be used in an expanded search if other than the default Roget. This is not an exposed parameter.

isGlobalSearcher As Boolean = False – Set to true if the current user is a global searcher. This is not an exposed parameter.

NbrOfErrors As Integer = 0 –An internal counter that returns the number of errors encountered during archive. Note: does not pertain to this API. This is not an exposed parameter.

gSql As String = "" – This is an internal global parameter that is set by ECM to establish the current SQL statement that is to be executed. This is not an exposed parameter.

gErrMsg As String = "" – This is an internal parameter that is set by ECM to capture and present the last encountered error message. This is not an exposed parameter.

### EMAIL Search Parameters

MinRating As Integer = 0 to 2500

When a weighted search is provided, this limits the rows returned to be at least what is specified by this value.

**Controlled by:**

Public Sub setMinRating(ByVal SessionGuid$, ByVal vMinRating As Integer)

MyContentOnly As Boolean = False

When it is desired to search only in within the content of the currently logged in user, set this value to true. The search and returned data will be limited to only that owned by the current user.

**Controlled by:**

Public Sub setMyContentOnly(ByVal SessionGuid$, ByVal vMyContentOnly As Boolean)

MasterContentOnly As Boolean = False

When it is desired to search only for Master Content, set this value to true. The search and returned data will be limited to only that content set as Master Content.

**Controlled by:**

Public Sub setCkSearchMasterDocsOnly(ByVal SessionGuid$, ByVal vCkSearchMasterDocsOnly As Boolean)

CkLimitToLib As Boolean = False - This is a critical part of a reduction search. If set to ‘false’ a search will traverse the entire repository for correcponding emails or content. If set to true, the search will be limited to include only the data retured that is contained within the specified library.

**Controlled by:**

Public Sub setCkLimitToLib(ByVal SessionGuid$, ByVal vCkLimitToLib As Boolean)

CbLibrary As String = "" – Enter the library name that is to be searched in this parameter.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public Sub setLibrary(ByVal SessionGuid$, ByVal vCbLibrary As String)

TxtSubject As String = "" – Set part or all of an email’s subject to reduce the search to include only emails that contain the specified character string. Wild card searches are allowed using the standard SQL qild card search criteria of a percent sign.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CkWeighted As Boolean = False – set this parameter to true to to bring back a weighted search. Works the same for both content and email.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CkBusiness As Boolean = False - set this parameter to true to to execute a “free text” search. A cautionary word, please read the help on this capability carefully. It can absorb a machine in a heartbeat. Works the same for both content and email.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CkIncludeAttachments As Boolean = True - set this parameter to true to to execute a search on emails that includes all of the email attachments. It should be set to false only under conditions where one is cetain that attachments should not be searched – and, this is not often, if ever.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CkDays As Boolean = False - set this parameter to true to to execute a search on emails that limits the retuned search to emails created within the last number of “XX” days.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

TxtDays As String = "" – specifies the number of days to span for the email reduction search.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CbFromAddr As String = "" – This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CbFromName As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CbToAddr As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CbToName As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

RbMyEmails As Boolean = False - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CbFolderFilter As String = "" – this is the email FOLDER as defined under Outlook. By placing a name in this field, all returned data will be limited to the specified folder or part of a folder name.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CbCCaddr As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

TxtCCPhrase As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CDateSelection As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CkSentDate As Boolean = False - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

DtMailDateStart As Boolean = False

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

DtMailDateEnd As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CbDateSelection As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CkCreationDate As Boolean = False - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

DtMailDateStart As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CkReceivedTime As Boolean = False - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

### Content Search Parameters

MinRating As Integer = 0 – Please refer to prior definition.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

MyContentOnly As Boolean = False – Please refer to prior definition.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

MasterContentOnly As Boolean = False – Please refer to prior definition.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

LibraryName As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

BIncludeLibraryFilesInSearch As Boolean = False - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CkSearchMasterDocsOnly As Boolean = False - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CbFilename As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CbFileTypes As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CbDirs As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CbMeta1 As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

TxtMetaSearch1 As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CbMeta2 As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

TxtMetaSearch2 As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CbMeta3 As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

TxtMetaSearch3 As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CbMeta4 As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

TxtMetaSearch4 As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

DtCreateDateStart As String = "" – This is the starting date when searching on a CREATE DATE for a document. It is a date value and when searching for a document by create date it works in conjunction with cbEvalCreateTime being set to one of the following: OFF, After, Before, Between, Not Between, On. This forms a reduction search. Data placed in this field will reduce the returned data to contain only content created on or before this date.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

DtCreateDateEnd As String = "" - This is the ending date when searching on a CREATE DATE for a document. It is a date value and when searching for a document by create date it works in conjunction with cbEvalCreateTime being set to one of the following: OFF, After, Before, Between, Not Between, On. This forms a reduction search. Data placed in this field will reduce the returned data to contain only content created on or before this date.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

DtLastWriteEnd As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

DtLastWriteStart As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CbEvalCreateTime As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

CbEvalWriteTime As String = "" - This forms a reduction search. Data placed in this field will reduce the returned data to contain only data that contains the data specified within this field.

**Controlled by: \*\* This item will be completely expanded upon request.**

Public

# Programming Examples:

### Search for a document by create date between two dates

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim SearchText As String = txtSearch.Text

If SearchText.Trim.Length = 0 Then

MsgBox("Please fill in the search line, returning.")

Return

End If

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Allowed Values

' OFF

' After

' Before

' Between

' Not Between

' On

SVC.setCbEvalCreateTime(SessionGuid, "Between")

SVC.setDtCreateDateEnd(SessionGuid, "01-01-2005")

SVC.setDtCreateDateEnd(SessionGuid, "12-31-2005")

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim dsTest As New DataSet

dsTest = SVC.SearchEmail(SessionGuid, SearchText)

If dsTest.Tables(0).Rows.Count > 0 Then

SB.Text = "Returned " + dsTest.Tables(0).Rows.Count.ToString + " rows."

Me.DataGridView1.DataSource = dsTest.Tables(0)

End If

SVC = Nothing

### Search for a document by create date before a date

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim SearchText As String = txtSearch.Text

If SearchText.Trim.Length = 0 Then

MsgBox("Please fill in the search line, returning.")

Return

End If

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Allowed Values

' OFF

' After

' Before

' Between

' Not Between

' On

SVC.setCbEvalCreateTime(SessionGuid, "Before")

SVC.setDtCreateDateEnd(SessionGuid, "01-01-2005")

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim dsTest As New DataSet

dsTest = SVC.SearchEmail(SessionGuid, SearchText)

If dsTest.Tables(0).Rows.Count > 0 Then

SB.Text = "Returned " + dsTest.Tables(0).Rows.Count.ToString + " rows."

Me.DataGridView1.DataSource = dsTest.Tables(0)

End If

SVC = Nothing

### Search for a document by create date ON a specific date

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim SearchText As String = txtSearch.Text

If SearchText.Trim.Length = 0 Then

MsgBox("Please fill in the search line, returning.")

Return

End If

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Allowed Values

' OFF

' After

' Before

' Between

' Not Between

' On

SVC.setCbEvalCreateTime(SessionGuid, "On")

SVC.setDtCreateDateEnd(SessionGuid, "01-01-2005")

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim dsTest As New DataSet

dsTest = SVC.SearchEmail(SessionGuid, SearchText)

If dsTest.Tables(0).Rows.Count > 0 Then

SB.Text = "Returned " + dsTest.Tables(0).Rows.Count.ToString + " rows."

Me.DataGridView1.DataSource = dsTest.Tables(0)

End If

SVC = Nothing

### Execute a Dynamic SQL Statement

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim MySql As String = "Select UserName, UserLoginID from Users order by UserName"

Dim EcmRepositoryConnStr$ = "Provider=SQLOLEDB;Data Source=DELLT100;Integrated Security=SSPI;Initial Catalog=ECM.Library"

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim B As Boolean = SVC.setConnStr(SessionGuid, EcmRepositoryConnStr$)

If B Then

MsgBox("Connection String successfully set.")

End If

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\* Search using the new connection string

Dim strXml$ = ""

strXml$ = SVC.getSqlDsXML(MySql, EcmRepositoryConnStr$)

MsgBox(strXml)

SVC = Nothing

### Restore an EMAIL

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim RestoreDirectory As String = "c:\temp"

Dim OverwriteDocumentIfExtsts As Boolean = True

Dim UserIdentifier As String = System.Net.Dns.GetHostName()

Dim Msg$ = "For testing, you will need to pass in the EMAIL Guid to this function" + vbCrLf

Msg$ += "Please look up a EMAIL guid and provide it here." + vbCrLf

Msg$ += "Note: There must be a C:\temp direcotry on this machine to proceed:" + vbCrLf

Dim EmailGuid As String = " "

InputBox(Msg, "EMAIL Guid")

Dim B As Boolean = SVC.restoreEmail(SessionGuid$, EmailGuid$, RestoreDirectory, OverwriteDocumentIfExtsts)

If B = True Then

MsgBox("EMAIL successfully restored to " + RestoreDirectory)

End If

SVC = Nothing

### Search for a document created after a specific date

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim SearchText As String = txtSearch.Text

If SearchText.Trim.Length = 0 Then

MsgBox("Please fill in the search line, returning.")

Return

End If

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Allowed Values

' OFF

' After

' Before

' Between

' Not Between

' On

SVC.setCbEvalCreateTime(SessionGuid, "After")

SVC.setDateCreateDateStart(SessionGuid, "01-01-2005")

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim dsTest As New DataSet

dsTest = SVC.SearchEmail(SessionGuid, SearchText)

If dsTest.Tables(0).Rows.Count > 0 Then

SB.Text = "Returned " + dsTest.Tables(0).Rows.Count.ToString + " rows."

Me.DataGridView1.DataSource = dsTest.Tables(0)

End If

SVC = Nothing

### Search for a document created between dates

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim SearchText As String = txtSearch.Text

If SearchText.Trim.Length = 0 Then

MsgBox("Please fill in the search line, returning.")

Return

End If

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Allowed Values

' OFF

' After

' Before

' Between

' Not Between

' On

SVC.setCbEvalCreateTime(SessionGuid, "Between")

SVC.setDateCreateDateStart(SessionGuid, "01-01-2005")

SVC.setDateCreateDateEnd(SessionGuid, "12-31-2005")

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim dsTest As New DataSet

dsTest = SVC.SearchEmail(SessionGuid, SearchText)

If dsTest.Tables(0).Rows.Count > 0 Then

SB.Text = "Returned " + dsTest.Tables(0).Rows.Count.ToString + " rows."

Me.DataGridView1.DataSource = dsTest.Tables(0)

End If

SVC = Nothing

### Search for Email after a specific SENT date

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim SearchText As String = txtSearch.Text

If SearchText.Trim.Length = 0 Then

MsgBox("Please fill in the search line, returning.")

Return

End If

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Allowed Values

' OFF

' After

' Before

' Between

' Not Between

' On

SVC.setCbEvalCreateTime(SessionGuid, "After")

SVC.setDateCreateDateStart(SessionGuid, "01-01-2005")

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

‘\*\* One of the folowing three items

‘MUST be selected.

‘

'\* Set search to run against SENT date

SVC.setCkSentDate(SessionGuid, True)

''\* Set search to run against CREATE date

'SVC.setCkCreationDate(SessionGuid, True)

''\* Set search to run against RECEIVED date

'SVC.setCkReceivedTime(SessionGuid, True)

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim dsTest As New DataSet

dsTest = SVC.SearchEmail(SessionGuid, SearchText)

If dsTest.Tables(0).Rows.Count > 0 Then

SB.Text = "Returned " + dsTest.Tables(0).Rows.Count.ToString + " rows."

Me.DataGridView1.DataSource = dsTest.Tables(0)

End If

SVC = Nothing

### Search for Email after a specific RECEIVED date

Dim SessionGuid$ = Guid.NewGuid.ToString

Dim SVC As ECMSVC.Service1Soap = New ECMSVC.Service1SoapClient

Dim SearchText As String = txtSearch.Text

If SearchText.Trim.Length = 0 Then

MsgBox("Please fill in the search line, returning.")

Return

End If

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Allowed Values

' OFF

' After

' Before

' Between

' Not Between

' On

SVC.setCbEvalCreateTime(SessionGuid, "After")

SVC.setDateCreateDateStart(SessionGuid, "01-01-2005")

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\* Set search to run against SENT date

'SVC.setCkSentDate(SessionGuid, True)

''\* Set search to run against CREATE date

'SVC.setCkCreationDate(SessionGuid, True)

''\* Set search to run against RECEIVED date

SVC.setCkReceivedTime(SessionGuid, True)

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim dsTest As New DataSet

dsTest = SVC.SearchEmail(SessionGuid, SearchText)

If dsTest.Tables(0).Rows.Count > 0 Then

SB.Text = "Returned " + dsTest.Tables(0).Rows.Count.ToString + " rows."

Me.DataGridView1.DataSource = dsTest.Tables(0)

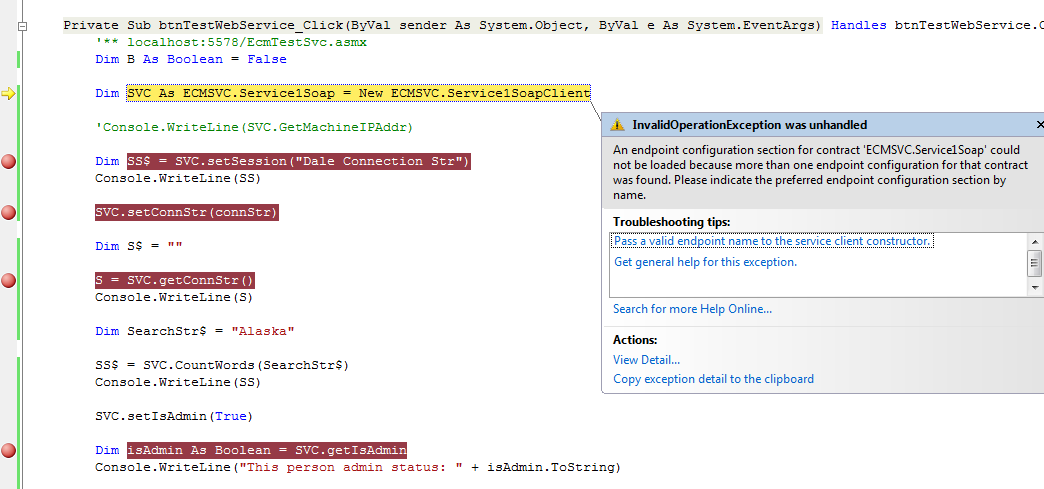
End If

SVC = Nothing

# Endpoint configuration error:

This is an interesting error, but very easy to correct.

Step 1 – Open your app.config file.





Step 2 – go to the Client, Endpoint section(usually at the end)

Step 3 – you will see multiple entries for endpoint address.



Step 4 – remove all but the last one.

Step 5 – recompile your program.

# ECM Web Services Source Code:

**Copyright © ECM Library, LLC, July 2009, all rights reserved.**

‘‘‘ <summary>

''' getConnectionString reads the default database connection string from the WEB CONFIG file.

''' It looks for the named pair identiied by "EcmLibraryCS".

''' </summary>

''' <returns></returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function getConnectionString() As String

Try

CS = System.Configuration.ConfigurationManager.ConnectionStrings("EcmLibraryCS").ToString

Return CS

Catch ex As Exception

Return "FAILED"

End Try

End Function

‘‘‘ <summary>

''' Generates the SQL that will perform the search based on the passed in Search String.

''' </summary>

''' <param name="SessionGuid"></param>

''' <param name="SearchString">The search string in ECM syntax</param>

''' <returns></returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function genDocSearchSql(ByVal SessionGuid$, ByVal SearchString$) As String

'Dim GV As New clsGlobalVar(SessionGuid)

Try

If SearchString.Trim.Length = 0 Then

SearchString$ = GV.getSearchString(SessionGuid$)

Else

GV.setSearchString(SessionGuid$, SearchString$)

End If

If SearchString.Trim.Length = 0 Then

Return "ERROR: No search criteria supplied."

End If

Dim CkWeighted = GV.getCkWeighted(SessionGuid$)

Dim txtThesaurus$ = GV.getThesaurus(SessionGuid$)

Dim cbThesaurusText$ = GV.getThesaurusWords(SessionGuid$)

Dim ckLimitToLib As Boolean = GV.getCkLimitToLib(SessionGuid$)

Dim LibraryName$ = GV.getLibraryName(SessionGuid$)

Dim LimitToExistingRows = GV.getCkLimitToExistingRows(SessionGuid$)

Dim SQLDOC As New clsSqlGenContent(SessionGuid$)

SQLDOC.pSessionGuid = SessionGuid$

Dim S$ = SQLDOC.GenDocSearchSql(SearchString$)

SQLDOC = Nothing

''GV = Nothing

Return S

Catch ex As Exception

''GV = Nothing

Return "ERROR - genDocSearchSql: " + ex.Message

End Try

End Function

‘‘‘ <summary>

''' Generates the SQL that will perform the search based on the passed in Search String.

''' </summary>

''' <param name="SessionGuid"></param>

''' <param name="SearchString">The search string in ECM syntax</param>

''' <returns></returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function genEmailSearchSql(ByVal SessionGuid$, ByVal SearchString$) As String

'Dim GV As New clsGlobalVar(SessionGuid)

Try

If SearchString.Trim.Length = 0 Then

SearchString$ = GV.getSearchString(SessionGuid$)

Else

GV.setSearchString(SessionGuid$, SearchString$)

End If

If SearchString.Trim.Length = 0 Then

Return "ERROR: No search criteria supplied."

End If

'Dim LimitToExistingRows = GV.GetCkLimitToExistingRows(SessionGuid$)

Dim SQLEMAIL As New clsSqlGenEmail(SessionGuid$)

SQLEMAIL.pSessionGuid = SessionGuid$

Dim S$ = SQLEMAIL.GenEmailSearchSql(SearchString$)

SQLEMAIL = Nothing

''GV = Nothing

Return S

Catch ex As Exception

''GV = Nothing

Return "ERROR - genEmailSearchSql: " + ex.Message

End Try

End Function

‘‘‘ <summary>

''' Performs a search on EMAIL based on selected setup parameters and a passed in Search String.

''' </summary>

''' <param name="SearchString">This is the SEARCH string that is passed in by the calling application. It must be in PROPER syntactical form.</param>

''' <param name="LimitToExistingRows">If set to TRUE, then only rows from the last search will be included into the current search.</param>

''' <param name="LimitSearchToLibrary">If set to TRUE, only content within the specified Library will be searched.</param>

''' <param name="LibraryName">If the above is true, then a library name must be supplied. This is where you provide the name of the library to be searched.</param>

''' <returns>a .Net Dataset</returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function SearchEmail(ByVal SessionGuid$, ByVal SearchString$) As DataSet

'SessionGuid$ = Guid.NewGuid.ToString

'Dim GV As New clsGlobalVar(SessionGuid)

updateIPAddr(SessionGuid$, 0)

updateIPAddr(SessionGuid$, 2)

GetHostName()

If Not MachineRegistered() Then

Return Nothing

End If

If Not MachineRegistered() Then

Return Nothing

End If

GV.pSessionGuid = SessionGuid$

'EcmLibraryCS = Web Config connection string

If SearchString$ = "" Then

Return Nothing

End If

If CS.Length = 0 Then

CS = System.Configuration.ConfigurationManager.ConnectionStrings("EcmLibraryCS").ToString

End If

GV.setSearchString(SessionGuid, SearchString$)

Dim LimitToExistingRows As Boolean = GV.getCkLimitToExistingRows(SessionGuid$)

Dim LimitSearchToLibrary As Boolean = GV.getCkLimitToLib(SessionGuid$)

Dim LibraryName$ = GV.getLibraryName(SessionGuid$)

Dim TgtThesaurus$ = GV.getTargetThesarus(SessionGuid$)

Dim ThesaurusWords$ = GV.getThesarusWords(SessionGuid$)

Dim DS As DataSet

Dim SEARCH As New clsSvcSearch(SessionGuid$)

SEARCH.pSessionGuid = SessionGuid$

SEARCH.SetConnectionString(CS)

'DS = SEARCH.SearchEmail(SearchString$, LimitToExistingRows, TgtThesaurus$, ThesaurusWords$, LimitSearchToLibrary, LibraryName$)

DS = SEARCH.SearchEmail()

SEARCH = Nothing

''GV = Nothing

Dim sRows$ = DS.Tables(0).Rows.Count.ToString

Console.WriteLine("Returned " + DS.Tables(0).Rows.Count.ToString + " rows of data.")

Return DS

End Function

‘‘‘ <summary>

''' Performs a search on EMAIL based on selected setup parameters and a passed in Search String.

''' </summary>

''' <param name="SessionGuid"></param>

''' <param name="SearchString">An ECM Search string in proper syntax.</param>

''' <returns>A well-formed XML dataset in the form of a string</returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function SearchEmailXML(ByVal SessionGuid$, ByVal SearchString$) As String

'Dim GV As New clsGlobalVar(SessionGuid)

updateIPAddr(SessionGuid$, 0)

updateIPAddr(SessionGuid$, 2)

GetHostName()

If Not MachineRegistered() Then

Return Nothing

End If

'EcmLibraryCS = Web Config connection string

If SearchString$ = "" Then

Return Nothing

End If

If CS.Length = 0 Then

CS = System.Configuration.ConfigurationManager.ConnectionStrings("EcmLibraryCS").ToString

End If

GV.setSearchString(SessionGuid, SearchString$)

Dim LimitToExistingRows As Boolean = GV.getCkLimitToExistingRows(SessionGuid$)

Dim LimitSearchToLibrary As Boolean = GV.getCkLimitToLib(SessionGuid$)

Dim LibraryName$ = GV.getLibraryName(SessionGuid$)

Dim TgtThesaurus$ = GV.getTargetThesarus(SessionGuid$)

Dim ThesaurusWords$ = GV.getThesarusWords(SessionGuid$)

Dim XML As String

Dim SEARCH As New clsSvcSearch(SessionGuid$)

SEARCH.pSessionGuid = SessionGuid$

SEARCH.SetConnectionString(CS)

'XML = SEARCH.SearchEmail(SearchString$, LimitToExistingRows, TgtThesaurus$, ThesaurusWords$, LimitSearchToLibrary, LibraryName$).GetXml

XML = SEARCH.SearchEmail().GetXml

SEARCH = Nothing

'GV = Nothing

Return XML

End Function

‘‘‘ <summary>

''' Performs a search on SOURCE/Documents based on selected setup parameters and a passed in Search String.

''' </summary>

''' <param name="SessionGuid"></param>

''' <param name="SearchString">An ECM search string in proper syntax</param>

''' <returns>A .Net dataset</returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function SearchDocuments(ByVal SessionGuid$, ByVal SearchString$) As DataSet

'Dim GV As New clsGlobalVar(SessionGuid)

Dim UserName$ = getUserName()

updateIPAddr(SessionGuid$, 0)

updateIPAddr(SessionGuid$, 2)

GetHostName()

If Not MachineRegistered() Then

Return Nothing

End If

'EcmLibraryCS = Web Config connection string

If SearchString$ = "" Then

Return Nothing

End If

If CS.Length = 0 Then

CS = System.Configuration.ConfigurationManager.ConnectionStrings("EcmLibraryCS").ToString

End If

GV.setSearchString(SessionGuid, SearchString$)

Dim CkWeighted As Boolean = GV.getCkWeighted(SessionGuid)

Dim LimitToExistingRows As Boolean = GV.getCkLimitToExistingRows(SessionGuid$)

Dim LimitSearchToLibrary As Boolean = GV.getCkLimitToLib(SessionGuid$)

Dim LibraryName$ = GV.getLibraryName(SessionGuid$)

Dim TgtThesaurus$ = GV.getTargetThesarus(SessionGuid$)

Dim ThesaurusWords$ = GV.getThesarusWords(SessionGuid$)

Dim DS As DataSet

Dim SEARCH As New clsSvcSearch(SessionGuid$)

SEARCH.pSessionGuid = SessionGuid$

SEARCH.SetConnectionString(CS)

DS = SEARCH.SearchDocuments(SearchString, CkWeighted, LimitToExistingRows, LimitSearchToLibrary, LibraryName$, TgtThesaurus$, ThesaurusWords$)

SEARCH = Nothing

'GV = Nothing

Return DS

End Function

‘‘‘ <summary>

''' Performs a search on SOURCE/Documents based on selected setup parameters and a passed in Search String.

''' </summary>

''' <param name="SessionGuid">An ECM search string in proper syntax</param>

''' <param name="SearchString"></param>

''' <returns>A well-formed XML dataset in the form of a string</returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function SearchDocumentsXML(ByVal SessionGuid$, ByVal SearchString$) As String

'Dim GV As New clsGlobalVar(SessionGuid)

updateIPAddr(SessionGuid$, 0)

updateIPAddr(SessionGuid$, 2)

GetHostName()

If Not MachineRegistered() Then

Return Nothing

End If

'EcmLibraryCS = Web Config connection string

If SearchString$ = "" Then

Return Nothing

End If

If CS.Length = 0 Then

CS = System.Configuration.ConfigurationManager.ConnectionStrings("EcmLibraryCS").ToString

End If

GV.setSearchString(SessionGuid, SearchString$)

Dim CkWeighted As Boolean = GV.getCkWeighted(SessionGuid)

Dim LimitToExistingRows As Boolean = GV.getCkLimitToExistingRows(SessionGuid$)

Dim LimitSearchToLibrary As Boolean = GV.getCkLimitToLib(SessionGuid$)

Dim LibraryName$ = GV.getLibraryName(SessionGuid$)

Dim TgtThesaurus$ = GV.getTargetThesarus(SessionGuid$)

Dim ThesaurusWords$ = GV.getThesarusWords(SessionGuid$)

Dim DS As DataSet

Dim XML As String = ""

Dim SEARCH As New clsSvcSearch(SessionGuid$)

SEARCH.pSessionGuid = SessionGuid$

SEARCH.SetConnectionString(CS)

XML = SEARCH.SearchDocuments(SearchString, CkWeighted, LimitToExistingRows, LimitSearchToLibrary, LibraryName$, TgtThesaurus$, ThesaurusWords$).GetXml

SEARCH = Nothing

'GV = Nothing

Return XML

End Function

‘‘‘ <summary>

''' Sets the current ECM Repository connection string to the passed in Connection String.

''' </summary>

''' <param name="SessionGuid"></param>

''' <param name="ConnStr">Well formed SQL DB Connection string.</param>

''' <returns>True id succesful, false if not</returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function setConnStr(ByVal SessionGuid$, ByVal ConnStr$) As Boolean

'Dim GV As New clsGlobalVar(SessionGuid)

Try

Application("appConnstr") = ConnStr

GV.setConnStr(SessionGuid$, ConnStr)

CS = ConnStr

Return True

Catch ex As Exception

Return False

End Try

End Function

‘‘‘ <summary>

''' Not used currently. An internal function.

''' </summary>

''' <param name="SessionVarName">Session Variable Name.</param>

''' <param name="tVar">The value to assign to the session variable.</param>

''' <returns></returns>

''' <remarks></remarks>

#### <WebMethod()> Private Function setSession(ByVal SessionVarName$, ByVal tVar$) As String

Dim S$ = ""

Try

'Session("XXX") = tVar

System.Web.HttpContext.Current.Session(SessionVarName$) = tVar

S = "Good"

Catch ex As Exception

S = ex.Message + " Could nit set " + SessionVarName$ + " to " + tVar + "."

WSERR.WriteToWebSvcLog(100, "setSession", S, CS)

End Try

Return S

End Function

‘‘‘ <summary>

''' Not currently used.

''' Returns the connection string supplied within the app.config file.

''' </summary>

''' <returns>String variable</returns>

''' <remarks></remarks>

#### <WebMethod()> Private Function getConnStr() As String

Try

Return Application("appConnstr").ToString

Catch ex As Exception

Return ""

End Try

'Return System.Web.HttpContext.Current.Session("XX")

End Function

‘‘‘ <summary>

''' Pass in a proper select statement and a count of rows returned will be returned.

''' </summary>

''' <param name="SelectSql">The sql to execute</param>

''' <param name="ConnStr"></param>

''' <returns>count of rowsgenerated by SQL statement - returns a -1 if FAILED.</returns>

''' <remarks></remarks>

#### <WebMethod()> Private Function iCount(ByVal SelectSql As String, ByVal ConnStr$) As Integer

Try

Dim S$ = SelectSql

'CS = System.Configuration.ConfigurationManager.ConnectionStrings("EcmLibraryCS").ToString

'Dim CS = ConnStr

Dim CON As New SqlConnection(ConnStr$)

Dim daCount As SqlDataAdapter = New SqlDataAdapter(S, ConnStr$)

'\*\* Create the client side dataset

Dim DS As New DataSet

'\*\* Explicitly open the connection to allow explicit closing.

CON.Open()

'\*\* Fill the dataset

daCount.Fill(DS, "Email")

Dim RecordCount As Integer = DS.Tables(0).Rows.Count

'\*\* Explicitly close the connection - do not wait for garbage collection.

CON.Close()

DS = Nothing

'\*\* Return the DataSet to the client.

Return RecordCount

Catch ex As Exception

WriteErr(2500, "ws:iCount", getUserName, ex.Message + SelectSql)

End Try

Return -1

End Function

''' <summary>

‘‘‘ Get a count of the tokens contained withint the search line. It can be used to limit the complexity of certain searches based strictly on the count of search tokens.

‘‘‘ </summary>

‘‘‘ <param name="SearchString"></param>

‘‘‘ <returns>Int count of tokens</returns>

‘‘‘ <remarks></remarks>

#### <WebMethod()> Public Function CountSearchTokens(ByVal SearchString As String) As String

Try

Dim Words() As String = SearchString.Split(" ")

Return "This Search String contains " + Words.Length.ToString + " word(s)."

Catch ex As Exception

WriteErr(2500, "ws:CountSearchTokens", getUserName, ex.Message)

Return (-1)

End Try

End Function

''' <summary>

‘‘‘ Pass in a well formed SQL Statement and get back the corresponding data in the form of a SQL DATASET

‘‘‘ </summary>

‘‘‘ <param name="SelectSql">a well formed SQL Statement that returns data</param>

‘‘‘ <param name="ConnStr">Current repositoy connection string</param>

‘‘‘ <returns>SQL DATASET</returns>

‘‘‘ <remarks></remarks>

#### <WebMethod()> Public Function getSqlDS(ByVal SelectSql As String, ByVal ConnStr$) As DataSet

Try

Dim S$ = SelectSql

'CS = System.Configuration.ConfigurationManager.ConnectionStrings("EcmLibraryCS").ToString

Dim CON As New SqlConnection(ConnStr$)

Dim daCount As SqlDataAdapter = New SqlDataAdapter(S, ConnStr$)

'\*\* Create the client side dataset

Dim DS As New DataSet

'\*\* Explicitly open the connection to allow explicit closing.

CON.Open()

'\*\* Fill the dataset

daCount.Fill(DS, "email")

'\*\* Explicitly close the connection - do not wait for garbage collection.

CON.Close()

'\*\* Return the DataSet to the client.

Return DS

Catch ex As Exception

WriteErr(2500, "ws:getSqlDS", getUserName, ex.Message)

Dim S$ = ex.Message.ToString

Return Nothing

End Try

End Function

''' <summary>

‘‘‘ Pass in a well formed SQL statement that returns data and get back the count of rows fetched by that statement.

‘‘‘ </summary>

‘‘‘ <param name="SelectSql">SQL Statement</param>

‘‘‘ <param name="ConnStr">repository connection string</param>

‘‘‘ <returns>integer number of rows</returns>

‘‘‘ <remarks></remarks>

#### <WebMethod()> Public Function getDataSetCount(ByVal SelectSql As String, ByVal ConnStr$) As Integer

Try

Dim S$ = SelectSql

'ConnStr$ = System.Configuration.ConfigurationManager.ConnectionStrings("EcmLibraryCS").ToString

Dim CON As New SqlConnection(ConnStr$)

Dim daCount As SqlDataAdapter = New SqlDataAdapter(S, ConnStr$)

'\*\* Create the client side dataset

Dim DS As New DataSet

'\*\* Explicitly open the connection to allow explicit closing.

CON.Open()

'\*\* Fill the dataset

daCount.Fill(DS, "Email")

Dim RecordCount As Integer = DS.Tables(0).Rows.Count

'\*\* Explicitly close the connection - do not wait for garbage collection.

CON.Close()

'\*\* Return the DataSet to the client.

Return RecordCount

Catch ex As Exception

WriteErr(2500, "ws:getDataSetCount", getUserName, ex.Message)

Return -1

End Try

End Function

''' <summary>

‘‘‘ Pass in a well formed SQL Statement and get back the corresponding data in the form of an XML dataset as a string

‘‘‘ </summary>

‘‘‘ <param name="SelectSql">a well formed SQL Statement </param>

‘‘‘ <param name="ConnStr">repository connection string</param>

‘‘‘ <returns>XML dataset as a string</returns>

‘‘‘ <remarks></remarks>

#### <WebMethod()> Public Function getSqlDsXML(ByVal SelectSql As String, ByVal ConnStr$) As String

'Try

Try

Dim XmlData$ = ""

Dim S$ = SelectSql

'CS = System.Configuration.ConfigurationManager.ConnectionStrings("EcmLibraryCS").ToString

Dim CON As New SqlConnection(ConnStr$)

Dim daCount As SqlDataAdapter = New SqlDataAdapter(S, ConnStr$)

'\*\* Create the client side dataset

Dim DS As New DataSet

'\*\* Explicitly open the connection to allow explicit closing.

CON.Open()

'\*\* Fill the dataset

daCount.Fill(DS, "temp")

XmlData = DS.GetXml

DS.Dispose()

DS = Nothing

'\*\* Explicitly close the connection - do not wait for garbage collection.

CON.Close()

'\*\* Return the DataSet to the client.

Return XmlData

Catch ex As Exception

WriteErr(2500, "ws:getSqlDsXML", getUserName, ex.Message)

Return Nothing

End Try

End Function

‘‘‘ <summary>

''' Pass in a table name and get the number of records it contains.

''' </summary>

''' <param name="Tbl">The Table Name</param>

''' <returns>count of records - integer</returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function iCountRecordsInTable(ByVal Tbl As String) As Integer

Dim S$ = ""

Dim iCnt As Integer = 0

Try

Dim connStr$ = Me.getConnectionString

S$ = "Select count(\*) from " + Tbl

Dim CON As New SqlConnection(connStr$)

Dim daCount As SqlDataAdapter = New SqlDataAdapter(S, connStr$)

'\*\* Create the client side dataset

Dim DS As New DataSet

'\*\* Explicitly open the connection to allow explicit closing.

CON.Open()

'\*\* Fill the dataset

daCount.Fill(DS, Tbl)

'\*\* get the target data

If DS.Tables(0).Rows.Count > 0 Then

iCnt = DS.Tables(0).Rows(0).Item(0)

Else

iCnt = 0

End If

'\*\* Explicitly close the connection - do not wait for garbage collection.

CON.Close()

'\*\* Return the DataSet to the client.

Return iCnt.ToString

'Catch ex As Exception

''WriteToWebSvcLog("iCountRecordsInTable: " + ex.Message)

'S = "error"

'End Try

Catch ex As Exception

WriteErr(2500, "ws:iCountRecordsInTable", getUserName, ex.Message)

iCnt = -1

End Try

Return iCnt

End Function

‘‘‘ <summary>

''' Using the passed in Source Guid, retrieve the length of the archived content.

''' </summary>

''' <param name="SourceGuid">This is the GUID of the archived content.</param>

''' <returns>Double</returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function getSourceLength(ByVal SourceGuid$) As Double

Try

Dim DB As New clsDatabase(SourceGuid$)

Dim S$ = "select filelength from DataSource where SourceGuid = '" + SourceGuid$ + "'"

Dim FileLength As Double = DB.GetFileLength(SourceGuid$)

DB = Nothing

Return FileLength

Catch ex As Exception

WriteErr(2500, "ws:getSourceLength", getUserName, ex.Message)

End Try

End Function

‘‘‘ <summary>

''' Using the passed in Source Guid, retrieve the binary image of the stored content.

''' </summary>

''' <param name="SourceGuid">This is the GUID of the archived content.</param>

''' <param name="SourceLength">Length of the document to retrieve.In a 32 bit system limited by the sizze of the integer datatype.</param>

''' <returns>A binary byte array</returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function getSourceImage(ByVal SourceGuid As String, ByVal SourceLength As Integer) As Byte()

Try

Dim DB As New clsDatabase(SourceGuid$)

Dim bArray() As Byte

bArray = DB.fetchImageSource(SourceGuid$, "content", SourceLength)

DB = Nothing

Return bArray

Catch ex As Exception

WriteErr(2500, "ws:getSourceImage", getUserName, ex.Message)

Return Nothing

End Try

End Function

‘‘‘ <summary>

''' Using the passed in Email Guid, retrieve the binary image of the stored email.

''' </summary>

''' <param name="EmailGuid">This is the GUID of the archived email.</param>

''' <param name="SourceLength">Length of the document to retrieve.In a 32 bit system limited by the sizze of the integer datatype.</param>

''' <returns>A binary byte array</returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function getEmailImage(ByVal EmailGuid$, ByVal SourceLength As Integer) As Byte()

Try

Dim DB As New clsDatabase(EmailGuid$)

Dim bArray() As Byte

bArray = DB.fetchImageSource(EmailGuid$, "email", SourceLength)

DB = Nothing

Return bArray

Catch ex As Exception

WriteErr(2500, "ws:getEmailImage", getUserName, ex.Message)

End Try

Return Nothing

End Function

‘‘‘ <summary>

''' Restore a specific Content Item (document) from the repository DataSource Table."

''' </summary>

''' <param name="SessionGuid">The current Session Identifier</param>

''' <param name="SourceGuid">The unique GUID of the Content within the repository</param>

''' <param name="SavePath">The directory where to store the retrieved document. </param>

''' <param name="OverWrite">When true, existing documents will be overwritten, when false, exsiting documents will not be overwritten.</param>

''' <returns>True if successful, False if failed</returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function restoreDocument(ByVal SessionGuid$, ByVal SourceGuid$, ByVal SavePath$, ByVal OverWrite As Boolean) As Boolean

updateIPAddr(SessionGuid$, 0)

updateIPAddr(SessionGuid$, 2)

GetHostName()

If Not MachineRegistered() Then

Return Nothing

End If

Try

Dim bArray() As Byte = Nothing

Dim SourceLength As Integer = 0

Dim B As Boolean = False

Dim DB As New clsDatabase(SessionGuid$)

Dim SourceName$ = DB.GetSourceName(SourceGuid$)

Dim FQN$ = SavePath

Dim CH$ = Mid(FQN$, FQN$.Trim.Length, 1)

If CH.Equals("\") Then

Else

FQN$ = FQN$ + "\"

End If

FQN$ = SavePath + SourceName

B = DB.writeImageSourceDataFromDbWriteToFile(SourceGuid$, FQN$, OverWrite)

DB = Nothing

Return B

Catch ex As Exception

WriteErr(2500, "ws:restoreDocument", getUserName, ex.Message)

Console.WriteLine(ex.Message)

Return False

End Try

End Function

#### <WebMethod()> Public Function restoreEmail(ByVal SessionGuid$, ByVal EmailGuid$, ByVal SavePath$, ByVal OverWrite As Boolean) As Boolean

updateIPAddr(SessionGuid$, 0)

updateIPAddr(SessionGuid$, 2)

GetHostName()

If Not MachineRegistered() Then

Return Nothing

End If

Try

Dim bArray() As Byte = Nothing

Dim SourceLength As Integer = 0

Dim B As Boolean = False

Dim DB As New clsDatabase(SessionGuid$)

Dim Suffix$ = DB.GetEmailSuffix(EmailGuid$)

Dim CH$ = Mid(SavePath, SavePath.Trim.Length, 1)

If CH.Equals("\") Then

Else

SavePath = SavePath + "\"

End If

'FQN$ = SavePath + SourceName

B = DB.writeEmailFromDbToFile(EmailGuid$, SavePath$, Suffix$)

DB = Nothing

Return B

Catch ex As Exception

WriteErr(2500, "ws:restoreEmail", getUserName, ex.Message)

Console.WriteLine(ex.Message)

Return False

End Try

End Function

#### <WebMethod()> Public Function getEmailAttachmentImage(ByVal SessionGuid$, ByVal ContentType As String, ByRef SourceLength As Integer, ByVal AttachmentID As Integer) As Byte()

Try

Dim DB As New clsDatabase(SessionGuid$)

Dim bArray() As Byte

bArray = DB.fetchImageSource(SessionGuid$, "attachment", SourceLength, AttachmentID)

DB = Nothing

Return bArray

Catch ex As Exception

WriteErr(2500, "ws:getEmailAttachmentImage", getUserName, ex.Message)

End Try

Return Nothing

End Function

#### '<WebMethod()> 'Public Function WriteToWebSvcLog(ByVal Msg$) As Boolean

'Try

''Dim cPath As String = getTempEnvironDir()

'Dim cPath As String = "E:\temp"

'Dim TempFolder$ = cPath

'Dim M$ = Now.Month.ToString.Trim

'Dim D$ = Now.Day.ToString.Trim

'Dim Y$ = Now.Year.ToString.Trim

'Dim SerialNo$ = M + "." + D + "." + Y + "."

'Dim tFQN$ = TempFolder$ + "\ECMLibrary.Trace.ECMQry.Log." + SerialNo$ + "txt"

'' Create an instance of StreamWriter to write text to a file.

'Using sw As StreamWriter = New StreamWriter(tFQN, True)

'' Add some text to the file.

'sw.WriteLine(Now.ToString + ": " + Msg + vbCrLf)

'sw.Close()

'End Using

'Return True

'Catch ex As Exception

'Console.WriteLine(ex.Message)

'Return False

'End Try

'End Function

#### '<WebMethod()> 'Sub ResolveName(ByVal name As String)

'Try

'Dim hostInfo As IPHostEntry

'hostInfo = Dns.Resolve(name)

'Dim address As IPAddress

'For Each address In hostInfo.AddressList

'Console.WriteLine(address.ToString)

'Next address

'Catch ex As Exception

'Console.WriteLine(ex.ToString)

'End Try

'End Sub

‘‘‘ <summary>

''' Returns the current IP of the currenlty logged in user whether on the web, intranet or local machine.

''' </summary>

''' <returns>IP as string</returns>

''' <remarks></remarks>

#### <WebMethod()> Private Function getUserIP() As String

Try

Dim CustomerIP$ = ""

If (HttpContext.Current.Request.ServerVariables("HTTP\_VIA") <> "") Then

CustomerIP = HttpContext.Current.Request.ServerVariables("HTTP\_X\_FORWARDED\_FOR").ToString()

Else

CustomerIP = HttpContext.Current.Request.ServerVariables("REMOTE\_ADDR").ToString()

End If

Return CustomerIP

Catch ex As Exception

WriteErr(2500, "ws:getUserIP", getUserName, ex.Message)

End Try

Return Nothing

End Function

‘‘‘ <summary>

''' Determines if the currently logged in machine is registerd to the repository.

''' </summary>

''' <returns>true if machine is registered, false if not.</returns>

''' <remarks></remarks>

#### <WebMethod()> Private Function MachineRegistered() As Boolean

Try

Dim B As Boolean = False

Dim STRHOSTNAME As String = ""

Dim STRIPADDRESS As String = ""

'Dim HOST As System.Net.IPHostEntry

STRHOSTNAME = System.Net.Dns.GetHostName()

STRIPADDRESS = System.Net.Dns.GetHostEntry(STRHOSTNAME).HostName.ToString()

Dim DB As New clsDatabase(SessionGuid$)

Dim MandateSeatLogin$ = System.Configuration.ConfigurationManager.AppSettings("MandateSeatLogin").ToUpper

If MandateSeatLogin$.Equals("TRUE") Then

B = DB.isMachineRegistered(STRHOSTNAME)

Else

B = True

End If

DB = Nothing

Return B

Catch ex As Exception

WriteErr(2500, "ws:MachineRegistered", getUserName, ex.Message)

Return "ERROR MachineRegistered: " + ex.Message

End Try

End Function

‘‘‘ <summary>

''' Gets the host name of the current logged in session.

''' </summary>

''' <returns>Hostname as string</returns>

''' <remarks></remarks>

#### <WebMethod()> Private Function GetHostName() As String

Try

Dim STRHOSTNAME As String = ""

Dim STRIPADDRESS As String = ""

Dim HOST As System.Net.IPHostEntry

STRHOSTNAME = System.Net.Dns.GetHostName()

STRIPADDRESS = System.Net.Dns.GetHostEntry(STRHOSTNAME).HostName.ToString()

Dim DB As New clsDatabase(SessionGuid$)

Dim MandateSeatLogin$ = System.Configuration.ConfigurationManager.AppSettings("MandateSeatLogin").ToUpper

If MandateSeatLogin$.Equals("TRUE") Then

DB.AddMachine(STRHOSTNAME)

End If

DB = Nothing

Return STRHOSTNAME

Catch ex As Exception

WriteErr(2500, "ws:GetHostName", getUserName, ex.Message)

Return "ERROR GetHostName: " + ex.Message

End Try

End Function

‘‘‘ <summary>

''' Returns the IP address of the executing/calling machine.

''' </summary>

''' <param name="SessionGuid"></param>

''' <returns>IP Address if successful, NULL string is not</returns>

''' <remarks></remarks>

#### <WebMethod()> Private Function GetMachineIPAddr(ByVal SessionGuid$) As String

'Dim GV As New clsGlobalVar(SessionGuid)

Try

Dim STRHOSTNAME As String = ""

Dim STRIPADDRESS As String = ""

Dim HOST As System.Net.IPHostEntry

STRHOSTNAME = System.Net.Dns.GetHostName()

STRIPADDRESS = System.Net.Dns.GetHostEntry(STRHOSTNAME).HostName.ToString()

HOST = System.Net.Dns.GetHostEntry(STRHOSTNAME)

Dim IP As System.Net.IPAddress

For Each IP In HOST.AddressList

GV.setHost(SessionGuid$, HOST.ToString)

GV.setMachineIPAddr(SessionGuid$, IP.ToString)

Return IP.ToString()

Next

Catch ex As Exception

WriteErr(2500, "ws:GetMachineIPAddr", getUserName, ex.Message)

End Try

Return ""

End Function

‘‘‘ <summary>

''' Returns the IP address of the executing/calling machine/client.

''' </summary>

''' <param name="SessionGuid"></param>

''' <returns>IP if successful, NULL if not.</returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function GetClientIPAddr(ByVal SessionGuid$) As String

Try

Dim STRHOSTNAME As String = ""

Dim STRIPADDRESS As String = ""

Dim HOST As System.Net.IPHostEntry

STRHOSTNAME = System.Net.Dns.GetHostName()

STRIPADDRESS = System.Net.Dns.GetHostEntry(STRHOSTNAME).HostName.ToString()

GV.setHost(SessionGuid$, STRHOSTNAME)

GV.setMachineIPAddr(SessionGuid$, STRIPADDRESS)

HOST = System.Net.Dns.GetHostEntry(STRHOSTNAME)

Dim IP As System.Net.IPAddress

'ip = HttpContext.Current.Request.ServerVariables["REMOTE\_ADDR"].ToString()

For Each IP In HOST.AddressList

Return IP.ToString()

Next

Catch ex As Exception

WriteErr(2500, "ws:GetClientIPAddr", getUserName, ex.Message)

End Try

Return ""

End Function

‘‘‘ <summary>

''' Updates the repository with a tracking of the current IP address.

''' </summary>

''' <param name="SessionGuid"></param>

''' <param name="UpdateCode">Reserved for future use. Not currently used. ' 0 = add if new,

''' 1 = update access count,

''' 2 = update search count,

''' 2 = update access count and search count.</param>

''' <remarks></remarks>

#### <WebMethod()> Private Sub updateIPAddr(ByVal SessionGuid$, ByVal UpdateCode As Integer)

'Dim GV As New clsGlobalVar(SessionGuid)

' 0 = add if new

' 1 = update access count

' 2 = update search count

' 2 = update access count and search count

Try

Dim STRHOSTNAME As String = GetHostName()

Dim STRIPADDRESS As String = GetClientIPAddr(SessionGuid$)

Dim HOST As System.Net.IPHostEntry

STRHOSTNAME = System.Net.Dns.GetHostName()

STRIPADDRESS = System.Net.Dns.GetHostEntry(STRHOSTNAME).HostName.ToString()

GV.setHost(SessionGuid$, STRHOSTNAME)

GV.setMachineIPAddr(SessionGuid$, STRIPADDRESS)

HOST = System.Net.Dns.GetHostEntry(STRHOSTNAME)

Dim IP As System.Net.IPAddress

'ip = HttpContext.Current.Request.ServerVariables["REMOTE\_ADDR"].ToString()

Dim CurrIP$ = ""

Dim LastIP$ = ""

For Each IP In HOST.AddressList

IP.ToString()

CurrIP = IP.ToString

If isFourPartIP(CurrIP) Then

Exit For

Else

CurrIP = ""

End If

LastIP = IP.ToString()

Next

If CurrIP.Length = 0 And LastIP.Length > 0 Then

CurrIP = LastIP

End If

Dim DB As New clsDatabase(SessionGuid$)

DB.updateIp(STRHOSTNAME, CurrIP, UpdateCode)

DB = Nothing

Catch ex As Exception

WriteErr(2500, "ws:updateIPAddr", getUserName, ex.Message)

End Try

End Sub

‘‘‘ <summary>

''' Determines if an IP address is a four part address or not.

''' </summary>

''' <param name="IP"></param>

''' <returns>true if 4 part, fasle if not</returns>

''' <remarks></remarks>

Function isFourPartIP(ByVal IP$) As Boolean

Try

If InStr(IP, ":") > 0 Then

Return False

End If

Dim A$() = IP.Split(".")

If A.Length = 4 Then

Return True

Else

Return False

End If

Catch ex As Exception

WriteErr(2500, "ws:isFourPartIP", getUserName, ex.Message)

End Try

Return False

End Function

‘‘‘ <summary>

''' Sets the search string.

''' </summary>

''' <param name="SessionGuid"></param>

''' <param name="SearchString"></param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setSearchString(ByVal SessionGuid$, ByVal SearchString As String)

GV.setSearchString(SessionGuid$, SearchString)

End Sub

‘‘‘ <summary>

''' Returns the current search string.

''' </summary>

''' <param name="SessionGuid"></param>

''' <returns>String</returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function getSearchString(ByVal SessionGuid$) As String

Return GV.GetSearchString(SessionGuid$)

End Function

‘‘‘ <summary>

''' Sets the parameter for including EMAIL Attachments in the search or not.

''' </summary>

''' <param name="IncludeAttachments"></param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setEmailIncludeAttachments(ByVal IncludeAttachments As Boolean)

ckIncludeAttachments = IncludeAttachments

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\* Operational Data

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

‘‘‘ <summary>

''' When called, will set the current user as an ADMIN or NOT

''' </summary>

''' <param name="SessionGuid"></param>

''' <param name="IsAdmin"></param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setIsAdmin(ByVal SessionGuid$, ByVal IsAdmin As Boolean)

GV.setIsAdmin(SessionGuid$, IsAdmin)

End Sub

‘‘‘ <summary>

''' Determines is current user is an administrator.

''' </summary>

''' <param name="SessionGuid"></param>

''' <returns>true if so, false if not</returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function getIsAdmin(ByVal SessionGuid$) As Boolean

Return GV.GetIsAdmin(SessionGuid$)

End Function

‘‘‘ <summary>

''' Sets the current user as global searcher or not.

''' </summary>

''' <param name="SessionGuid"></param>

''' <param name="IsGlobalSearcher"></param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setIsGlobalSearcher(ByVal SessionGuid$, ByVal IsGlobalSearcher As Boolean)

GV.setIsGlobalSearcher(SessionGuid$, IsGlobalSearcher)

End Sub

‘‘‘ <summary>

''' Determines if the current user is a global searcher or not.

''' </summary>

''' <param name="SessionGuid"></param>

''' <returns>true is so, false if not</returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function getIsGlobalSearcher(ByVal SessionGuid$) As Boolean

Return GV.getIsGlobalSearcher(SessionGuid)

End Function

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\* Database Class Parameters

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

‘‘‘ <summary>

''' Sets the current user GUID for the current session.

''' </summary>

''' <param name="SessionGuid"></param>

''' <param name="vCurrUserGuidID"></param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCurrUserGuidID(ByVal SessionGuid$, ByVal vCurrUserGuidID As String)

GV.setCurrUserGuidID(SessionGuid$, vCurrUserGuidID)

End Sub

‘‘‘ <summary>

''' Sets the search Master Content only indicator. Set to true to search only master content, fasle to search all.

''' </summary>

''' <param name="SessionGuid"></param>

''' <param name="vMasterContentOnly">boolean - true or false</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setMasterContentOnly(ByVal SessionGuid$, ByVal vMasterContentOnly As Boolean)

GV.setMasterContentOnly(SessionGuid$, vMasterContentOnly)

End Sub

‘‘‘ <summary>

''' Sets the search MY Content ONLY indicator. Set to true to search only the current user's content, fasle to search all.

''' </summary>

''' <param name="SessionGuid"></param>

''' <param name="vMyContentOnly">boolean - true or false</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setMyContentOnly(ByVal SessionGuid$, ByVal vMyContentOnly As Boolean)

GV.setMyContentOnly(SessionGuid$, vMyContentOnly)

End Sub

‘‘‘ <summary>

''' Programatically track the number of errors encounterd by a calling program.

''' </summary>

''' <param name="SessionGuid"></param>

''' <param name="vNbrOfErrors"></param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setNbrOfErrors(ByVal SessionGuid$, ByVal vNbrOfErrors As Integer)

GV.setNbrOfErrors(SessionGuid$, vNbrOfErrors)

End Sub

‘‘‘ <summary>

''' Set the current Session Guid to allow concurrent processing of many tasks by a "Session ID" / "Session Guid".

''' </summary>

''' <param name="SessionGuid">A generated GUID as a string</param>

''' <param name="vSql"></param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setSql(ByVal SessionGuid$, ByVal vSql As String)

GV.setSql(SessionGuid$, vSql)

End Sub

‘‘‘ <summary>

''' Set the text of an error message so that a call can be issued to the Error Subroutine at a later time.

''' Currently - not used.

''' </summary>

''' <param name="SessionGuid"></param>

''' <param name="vErrMsg">The error message text.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setErrMsg(ByVal SessionGuid$, ByVal vErrMsg As String)

GV.setErrMsg(SessionGuid$, vErrMsg)

End Sub

‘‘‘ <summary>

''' When performing a Weighted search, seeting this value causes only documents that have a weighting above this value to be reuturned.

''' </summary>

''' <param name="SessionGuid"></param>

''' <param name="vMinRating">A number</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setMinRating(ByVal SessionGuid$, ByVal vMinRating As Integer)

GV.setMinRating(SessionGuid$, vMinRating)

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\* Error tracking

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

‘‘‘ <summary>

''' Since web processes and at times, local process do not support entering data into the clipboard, we provided this method.

''' By calling this subroutine, data is stored in a reserved area attached to the current Session GUID and can be recalled

''' when needed just as with the clipboard.

''' </summary>

''' <param name="SessionGuid"></param>

''' <param name="vClipBoardData">Value to store into the clipboard.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setClipBoardData(ByVal SessionGuid$, ByVal vClipBoardData As String)

GV.setClipBoardData(SessionGuid$, vClipBoardData)

End Sub

‘‘‘ <summary>

''' Retrieves the data stored in the clipboard described in the function setClipBoardData.

''' </summary>

''' <param name="SessionGuid"></param>

''' <param name="vClipBoardData">Clipborard saved data.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub getClipBoardData(ByVal SessionGuid$, ByVal vClipBoardData As String)

GV.getClipBoardData(SessionGuid$)

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\* EMAIL Search Parameters

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

‘‘‘ <summary>

''' Set this to true to limit the search to a specific Library.

''' Must be used together with function "setLibrary"

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCkLimitToLib">When set to true, will cause the search to be limited to the specified library.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCkLimitToLib(ByVal SessionGuid$, ByVal vCkLimitToLib As Boolean)

GV.setEmailLimitToLib(SessionGuid$, vCkLimitToLib)

End Sub

‘‘‘ <summary>

''' Enter a library name here to reduce the search to a known library.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCbLibrary">Unique name of library to be searched.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setLibrary(ByVal SessionGuid$, ByVal vCbLibrary As String)

GV.setLibrary(SessionGuid$, vCbLibrary)

End Sub

‘‘‘ <summary>

''' Use this field to specify reduction criteria for the email search thru the SUBJECT of the email. Wild cards can be use such as %job offer% or %job offer or job offer% or just the phrase "job offer"

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vTxtSubject"></param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setEmailSubject(ByVal SessionGuid$, ByVal vTxtSubject As String)

GV.setEmailSubject(SessionGuid$, vTxtSubject)

End Sub

‘‘‘ <summary>

''' Set this to true to include the search WEIGHTS in the return data.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCkWeighted">Boolean expression such that true = on, false = off</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCkWeighted(ByVal SessionGuid$, ByVal vCkWeighted As Boolean)

GV.setEmailWeighted(SessionGuid$, vCkWeighted)

End Sub

‘‘‘ <summary>

''' Set this to true to use free text to search the content. Be advised this can cause a huge performance hit on the server.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCkBusiness">Boolean expression such that true = on, false = off</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCkBusiness(ByVal SessionGuid$, ByVal vCkBusiness As Boolean)

GV.setEmailBusiness(SessionGuid$, vCkBusiness)

End Sub

‘‘‘ <summary>

''' Set the search parameters in ECM Library syntax here.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vTxtSearch">The passed in search argument/parameters.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setTxtSearch(ByVal SessionGuid$, ByVal vTxtSearch As String)

GV.setSearchString(SessionGuid$, vTxtSearch)

End Sub

‘‘‘ <summary>

''' Set this to true to issue a reduction search limiting the search to only that content already contained within the

''' returned search items. Each search will continue to limit and search on the retured set of data.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCkLimitToExisting">Boolean expression such that true = on, false = off</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCkLimitToExisting(ByVal SessionGuid$, ByVal vCkLimitToExisting As Boolean)

GV.setCkLimitToExistingRows(SessionGuid$, vCkLimitToExisting)

End Sub

‘‘‘ <summary>

''' Deafault is true - turn to false to search only on emails and skip the attachments.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vckIncludeAttachments">Boolean expression such that true = on, false = off</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCkIncludeAttachments(ByVal SessionGuid$, ByVal vckIncludeAttachments As Boolean)

GV.setEmailIncludeAttachments(SessionGuid$, vckIncludeAttachments)

End Sub

‘‘‘ <summary>

''' Set to true so that the search will be limited only to content produced within the last XX number of days.

''' Use setTxtDays to set the number of days the search is to include.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCkDays">Boolean expression such that true = on, false = off</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCkDays(ByVal SessionGuid$, ByVal vCkDays As Boolean)

GV.setEmailDaysCk(SessionGuid$, vCkDays)

End Sub

‘‘‘ <summary>

''' A search can be limited to incloude content produced only within the last XX number of days.

''' Set the number of days the search is to include using this function. Be sure and set setCkDays

''' to true or the reduction search will not be executed.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vTxtDays">A number</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setTxtDays(ByVal SessionGuid$, ByVal vTxtDays As String)

GV.setEmailDaysNbr(SessionGuid$, vTxtDays)

End Sub

‘‘‘ <summary>

''' Use this field to specify reduction criteria for the email search thru the From Address of the email. Wild cards can be use such as %GeneralMotors% or %GeneralMotors or job GeneralMotors% or just the exact phrase "GeneralMotors"

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCbFromAddr">Free form text with wild card capability</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCbFromAddr(ByVal SessionGuid$, ByVal vCbFromAddr As String)

GV.setEmailFromAddr(SessionGuid$, vCbFromAddr)

End Sub

‘‘‘ <summary>

''' Use this field to specify reduction criteria for the email search thru the From NAME of the email. Wild cards can be use such as %William@% or %William or job Williams.com% or just the exact phrase "William@GMAC.com"

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCbFromName">Free form text with wild card capability</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCbFromName(ByVal SessionGuid$, ByVal vCbFromName As String)

GV.setEmailFromName(SessionGuid$, vCbFromName)

End Sub

‘‘‘ <summary>

''' Use this field to specify reduction criteria for the email search thru the TO ADDRESS of the email. Wild cards can be use such as %William@% or %William or job Williams.com% or just the exact phrase "William@GMAC.com"

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCbToAddr">Free form text with wild card capability</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCbToAddr(ByVal SessionGuid$, ByVal vCbToAddr As String)

GV.setEmailToAddr(SessionGuid$, vCbToAddr)

End Sub

‘‘‘ <summary>

''' Use this field to specify reduction criteria for the email search thru the TO NAME of the email. Wild cards can be use such as %William@% or %William or job Williams.com% or just the exact phrase "William@GMAC.com"

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCbToName">Free form text with wild card capability</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCbToName(ByVal SessionGuid$, ByVal vCbToName As String)

GV.setEmailToName(SessionGuid$, vCbToName)

End Sub

‘‘‘ <summary>

''' Set true to limit the search to ONLY the emails of the current searcher/user.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vRbMyEmails">Boolean value true to enable, false to disable</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setRbMyEmails(ByVal SessionGuid$, ByVal vRbMyEmails As Boolean)

GV.setMyEmailsOnly(SessionGuid$, vRbMyEmails)

End Sub

‘‘‘ <summary>

''' Use this field to specify reduction criteria for the email search to a specific EMAIL Folder. Wild cards can be use such as %INBOX@% or %INBOX or job INBOX% or just the exact phrase "INBOX"

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCbFolderFilter">Free form text with wild card capability</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCbFolderFilter(ByVal SessionGuid$, ByVal vCbFolderFilter As String)

GV.setEmailFolderFilter(SessionGuid$, vCbFolderFilter)

End Sub

‘‘‘ <summary>

''' Use this field to specify reduction criteria for the email search CC ADDRESSES of the email. Wild cards can be use such as %William@% or %William or job Williams.com% or just the exact phrase "William@GMAC.com"

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCbCCaddr">Free form text with wild card capability</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCbCCaddr(ByVal SessionGuid$, ByVal vCbCCaddr As String)

GV.setEmailCCaddr(SessionGuid$, vCbCCaddr)

End Sub

‘‘‘ <summary>

''' Use this field to specify reduction criteria for the email search CC ADDRESSES or any phrase contained within the CC field of the email. Wild cards can be use such as %William@% or %William or job Williams.com% or just the exact phrase "William@GMAC.com"

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vTxtCCPhrase">Free form text with wild card capability</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setTxtCCPhrase(ByVal SessionGuid$, ByVal vTxtCCPhrase As String)

GV.setEmailCCPhrase(SessionGuid$, vTxtCCPhrase)

End Sub

‘‘‘ <summary>

''' Not currently used. Internal function.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCbDateSelection"></param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setbDateSelection(ByVal SessionGuid$, ByVal vCbDateSelection As String)

GV.setEmailDateSelectionType(SessionGuid$, vCbDateSelection)

End Sub

‘‘‘ <summary>

''' Set this to true limit search to email sent date.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCkSentDate">A boolean parameter: set true forces the date reduction search to be performed against the EMAIL sent date.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCkSentDate(ByVal SessionGuid$, ByVal vCkSentDate As Boolean)

GV.setEmailSentDateCk(SessionGuid$, vCkSentDate)

End Sub

‘‘‘ <summary>

''' This is the end date for the email reduction search.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vDtMailDateEnd">Reduction search end date.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setDateMailDateEnd(ByVal SessionGuid$, ByVal vDtMailDateEnd As String)

GV.selEmailDateEndPeriod(SessionGuid$, vDtMailDateEnd)

End Sub

‘‘‘ <summary>

''' Set the date limiting parameter for an EMAIL search

''' </summary>

''' <param name="SessionGuid">Session ID</param>

''' <param name="vCbDateSelection">One of four values OFF,After,Before,Between,Not Between</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCbDateSelection(ByVal SessionGuid$, ByVal vCbDateSelection As String)

GV.setEmailDateSelectionType(SessionGuid$, vCbDateSelection)

End Sub

‘‘‘ <summary>

''' Set this to true limit search to email create date.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCkCreationDate">A boolean parameter: set true forces the date reduction search to be performed against the EMAIL create date.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCkCreationDate(ByVal SessionGuid$, ByVal vCkCreationDate As Boolean)

GV.setCreationDateCk(SessionGuid$, vCkCreationDate)

End Sub

‘‘‘ <summary>

''' This is the start date for the email reduction search.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vDtMailDateStart">Reduction search end date.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setDateMailDateStart(ByVal SessionGuid$, ByVal vDtMailDateStart As String)

GV.selEmailDateStartPeriod(SessionGuid$, vDtMailDateStart)

End Sub

‘‘‘ <summary>

''' Set this to true limit search to email received date.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCkReceivedTime">A boolean parameter: set true forces the date reduction search to be performed against the EMAIL received date.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCkReceivedTime(ByVal SessionGuid$, ByVal vCkReceivedTime As Boolean)

GV.setReceivedTimeCk(SessionGuid$, vCkReceivedTime)

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\* Content Search Parameters

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

‘‘‘ <summary>

''' setLibraryName - when a library name is supplied, the search will be limited to

''' the designated library.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vLibraryName">The uniqie name of the library to be searched.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setLibraryName(ByVal SessionGuid$, ByVal vLibraryName As String)

GV.setContentLibraryName(SessionGuid$, vLibraryName)

End Sub

‘‘‘ <summary>

''' Enable the content search to look in a specific library only.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vBIncludeLibraryFilesInSearch">A boolean parameter: set true to enable, false to disable.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setBIncludeLibraryFilesInSearch(ByVal SessionGuid$, ByVal vBIncludeLibraryFilesInSearch As Boolean)

GV.setContentIncludeLibraryFilesInSearchCk(SessionGuid$, vBIncludeLibraryFilesInSearch)

End Sub

‘‘‘ <summary>

''' Set search to look in Master documents only.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCkSearchMasterDocsOnly">A boolean parameter: set true to enable, false to disable.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCkSearchMasterDocsOnly(ByVal SessionGuid$, ByVal vCkSearchMasterDocsOnly As Boolean)

GV.setContentSearchMasterDocsOnlyCk(SessionGuid$, vCkSearchMasterDocsOnly)

End Sub

‘‘‘ <summary>

''' Use this field to specify reduction criteria for the content search to be limited to a specific File Name or any phrase contained within the File Name. Wild cards can be use such as %001% or %Contract.00% or 001.docx% or just the exact phrase "Contract.001.docx"

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCbFilename">Free form text with wild card capability</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCbFilename(ByVal SessionGuid$, ByVal vCbFilename As String)

GV.setContentFilename(SessionGuid$, vCbFilename)

End Sub

‘‘‘ <summary>

''' Use this field to specify reduction criteria for the content search to be limited to a specific File Types. Wild cards can be use such as %doc% or %.doc or docx% or just the exact phrase ".docx"

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCbFileTypes">Free form text with wild card capability</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCbFileTypes(ByVal SessionGuid$, ByVal vCbFileTypes As String)

GV.setContentFileType(SessionGuid$, vCbFileTypes)

End Sub

‘‘‘ <summary>

''' Use this field to specify reduction criteria for the content search to be limited to a specific DIRECTORY. Wild cards can be use such as %\corporate\documents% or %\corp or doeuments% or just the exact phrase "C:\corporate\documents"

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCbFileTypes">Free form text with wild card capability</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCbDirs(ByVal SessionGuid$, ByVal vCbDirs As String)

GV.SetContentDir(SessionGuid$, vCbDirs)

End Sub

‘‘‘ <summary>

''' Metadata tags are captured for each item of content. They canbe different depending upon the type of content. Four tags can be searched - this is where the Metadata Tag name is specified.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCbMeta1">The name of the metadata tag to search.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setMetaDataSearchKey1(ByVal SessionGuid$, ByVal vCbMeta1 As String)

GV.setContentMetadataKey1(SessionGuid$, vCbMeta1)

End Sub

‘‘‘ <summary>

''' When a metadata tag is specified to search (1 thru 4), this is where the corresponding value for that tag is specified.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vTxtMetaSearch1">Freeform test except in the case of numbers. Commas in numbers must be accurately represented.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setMetaDataSearchValue1(ByVal SessionGuid$, ByVal vTxtMetaSearch1 As String)

GV.setContentMetaSearchTxt1(SessionGuid$, vTxtMetaSearch1)

End Sub

‘‘‘ <summary>

''' Metadata tags are captured for each item of content. They canbe different depending upon the type of content. Four tags can be searched - this is where the Metadata Tag name is specified.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCbMeta2">The name of the metadata tag to search.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setMetaDataSearchKey2(ByVal SessionGuid$, ByVal vCbMeta2 As String)

GV.setContentMetadataKey2(SessionGuid$, vCbMeta2)

End Sub

‘‘‘ <summary>

''' When a metadata tag is specified to search (1 thru 4), this is where the corresponding value for that tag is specified.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vTxtMetaSearch2">Freeform test except in the case of numbers. Commas in numbers must be accurately represented.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setMetaDataSearchValue2(ByVal SessionGuid$, ByVal vTxtMetaSearch2 As String)

GV.setContentMetaSearchTxt2(SessionGuid$, vTxtMetaSearch2)

End Sub

‘‘‘ <summary>

''' Metadata tags are captured for each item of content. They canbe different depending upon the type of content. Four tags can be searched - this is where the Metadata Tag name is specified.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCbMeta3">The name of the metadata tag to search.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setMetaDataSearchKey3(ByVal SessionGuid$, ByVal vCbMeta3 As String)

GV.setContentMetadataKey3(SessionGuid$, vCbMeta3)

End Sub

‘‘‘ <summary>

''' When a metadata tag is specified to search (1 thru 4), this is where the corresponding value for that tag is specified.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vTxtMetaSearch3">Freeform test except in the case of numbers. Commas in numbers must be accurately represented.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setMetaDataSearchValue3(ByVal SessionGuid$, ByVal vTxtMetaSearch3 As String)

GV.setContentMetaSearchTxt3(SessionGuid$, vTxtMetaSearch3)

End Sub

‘‘‘ <summary>

''' Metadata tags are captured for each item of content. They canbe different depending upon the type of content. Four tags can be searched - this is where the Metadata Tag name is specified.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCbMeta4">The name of the metadata tag to search.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setMetaDataSearchKey4(ByVal SessionGuid$, ByVal vCbMeta4 As String)

GV.setContentMetadataKey4(SessionGuid$, vCbMeta4)

End Sub

‘‘‘ <summary>

''' When a metadata tag is specified to search (1 thru 4), this is where the corresponding value for that tag is specified.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vTxtMetaSearch4">Freeform test except in the case of numbers. Commas in numbers must be accurately represented.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setMetaDataSearchValue4(ByVal SessionGuid$, ByVal vTxtMetaSearch4 As String)

GV.setContentMetaSearchTxt4(SessionGuid$, vTxtMetaSearch4)

End Sub

‘‘‘ <summary>

''' The starting date used to establish a content reduction search on the create date of the document.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vDtCreateDateStart">A standard date. (e.g. "11/01/2009" "11-01-2009" or a variable (e.g. NOW.Tostring)</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setDateCreateDateStart(ByVal SessionGuid$, ByVal vDtCreateDateStart As String)

GV.setContentCreateDateStartDate(SessionGuid$, vDtCreateDateStart)

End Sub

‘‘‘ <summary>

''' The ending date used to establish a content reduction search on the create date of the document.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vDtCreateDateEnd">A standard date. (e.g. "11/01/2009" "11-01-2009" or a variable (e.g. NOW.Tostring)</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setDateCreateDateEnd(ByVal SessionGuid$, ByVal vDtCreateDateEnd As String)

GV.setContentCreateDateEndDate(SessionGuid$, vDtCreateDateEnd)

End Sub

‘‘‘ <summary>

''' The ending date used to establish a content reduction search on the LAST WRITE date of the document.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vDtLastWriteEnd">A standard date. (e.g. "11/01/2009" "11-01-2009" or a variable (e.g. NOW.Tostring)</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setDateLastWriteEnd(ByVal SessionGuid$, ByVal vDtLastWriteEnd As String)

GV.setContentLastWriteEndDate(SessionGuid$, vDtLastWriteEnd)

End Sub

‘‘‘ <summary>

''' The starting date used to establish a content reduction search on the LAST WRITE date of the document. If anything but "BETWEEN" is specified as the delimiting type, use this variable only.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vDtLastWriteStart"></param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setDateLastWriteStart(ByVal SessionGuid$, ByVal vDtLastWriteStart As String)

GV.setContentLastWriteStartDate(SessionGuid$, vDtLastWriteStart)

End Sub

‘‘‘ <summary>

''' Set the date limiting parameter for a CONTENT search where a documents CREATE date is to be evaluated.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCbEvalCreateTime">One of four values OFF,After,Before,Between,Not Between</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCbEvalCreateTime(ByVal SessionGuid$, ByVal vCbEvalCreateTime As String)

GV.setContentEvalCreateTimeCode(SessionGuid$, vCbEvalCreateTime)

End Sub

‘‘‘ <summary>

''' Set the date limiting parameter for a CONTENT search where a documents LAST WRITE date is to be evaluated.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vCbEvalWriteTime">One of four values OFF,After,Before,Between,Not Between</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCbEvalWriteTime(ByVal SessionGuid$, ByVal vCbEvalWriteTime As String)

GV.setContentEvalWriteTimeCode(SessionGuid$, vCbEvalWriteTime)

End Sub

‘‘‘ <summary>

''' There are times when it may be needed to provide the functionality to login as a differnet user from the same session or machine.

''' The new user login id can be set here. If it does not pass the "login" check, it will throw an error.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="vGurrUserLoginID">The New user login ID</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setGurrUserLoginID(ByVal SessionGuid$, ByVal vGurrUserLoginID As String)

GV.setGurrUserLoginID(SessionGuid$, vGurrUserLoginID)

End Sub

‘‘‘ <summary>

''' Write an error message to the PGMTRACE table in the repository.

''' </summary>

''' <param name="StmtID">A statement ID that is embedded within the calling application so that a refernece to where and what the error is can be established. </param>

''' <param name="PgmName">The name of the calling program, class, function</param>

''' <param name="UserID">The current userid</param>

''' <param name="ErrMsg">The message to record</param>

''' <remarks></remarks>

#### <WebMethod()> Sub WriteErr(ByVal StmtID As Integer, ByVal PgmName$, ByVal UserID$, ByVal ErrMsg As String)

Try

Dim ConnStr$ = getConnectionString()

Dim CON As New SqlConnection(ConnStr$)

Dim DS As New DataSet

CON.Open()

Dim DA As New SqlDataAdapter("SELECT [StmtID],[PgmName],[Stmt],[RowID],[CreateDate],[ConnectiveGuid],[UserID] FROM [PgmTrace] where 1 = 2", ConnStr)

Dim cmdBuilder As New SqlCommandBuilder(DA)

DA.Fill(DS, "PgmTrace")

Dim DT As DataTable = DS.Tables(0)

Dim NewRow As DataRow = DT.NewRow

NewRow("StmtID") = StmtID

NewRow("PgmName") = PgmName

NewRow("UserID") = UserID

NewRow("Stmt") = ErrMsg

NewRow("CreateDate") = Now

DT.Rows.Add(NewRow)

DA.Update(DT)

CON.Close()

Catch ex As Exception

Console.WriteLine(ex.Message)

Console.WriteLine(ex.Message)

End Try

End Sub

‘‘‘ <summary>

''' Internal function used to validate users with Active Directory. Not availble to coders except to get a refernce to a users current authentication status.

''' </summary>

''' <returns></returns>

''' <remarks></remarks>

#### <WebMethod()> Function CredentialInfo() As String

Try

Dim uriObj As New Uri("http://tempuri.org/")

Dim credentials As System.Net.ICredentials = System.Net.CredentialCache.DefaultCredentials

Dim credential As System.Net.NetworkCredential = credentials.GetCredential(uriObj, "Basic")

Dim UserID$ = credential.UserName.ToString

UserID = System.Security.Principal.WindowsIdentity.GetCurrent.Name

Return UserID

'Return System.Environment.UserName '\*\* Returns the global service user as defined on the server

Catch ex As Exception

Return ex.Message

End Try

End Function

‘‘‘ <summary>

''' Issues a check against the ECM license contained within the current repository to determine if it is valid or not.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <returns>True is valid, false if not. </returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function isLicenseValid(ByVal SessionGuid$) As Boolean

Dim I As Integer = -1

Dim DB As New clsDatabase(SessionGuid$)

Dim ErrMsg$ = ""

I = DB.LV(ErrMsg)

DB = Nothing

If I = 0 Then

Return True

Else

Return False

End If

End Function

‘‘‘ <summary>

''' This can be called by programmers to determine the state of the current ECM License.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <returns>0 - Good all the way,

''' 1 - No active license.,

''' 2 - Maximum number of work stations exceed.,

''' 3 - Maximum number of IP's and machine combinations exceed.,

''' 4 - Expired lease license.,

''' 5 - User licensing limit exceeded.

''' </returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function getLicenseStatus(ByVal SessionGuid$) As String

'\*\* 0 - Good all the way

'\*\* 1 - No active license.,

'\*\* 2 - Maximum number of work stations exceed.,

'\*\* 3 - Maximum number of IP's and machine combinations exceed.,

'\*\* 4 - Expired lease license.,

'\*\* 5 - User licensing limit exceeded.

Dim I As Integer = -1

Dim DB As New clsDatabase(SessionGuid$)

Dim ErrMsg$ = ""

I = DB.LV(ErrMsg)

DB = Nothing

Dim S$ = ""

If I = 0 Then

S = "License is good."

Else

S = "Failure Code: " + I.ToString + " - Failed License Test: " + ErrMsg

End If

Return S

End Function

‘‘‘ <summary>

''' Gets the currently logged in user's ID.

''' </summary>

''' <returns>The currently logged in user's ID.</returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function getUserName() As String

'Dim S$ = HttpContext.Current.Request.IsLocal

Return System.Environment.UserName

End Function

‘‘‘ <summary>

''' Gets the currently logged in user's domain name.

''' </summary>

''' <returns>logged in user's domain name</returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function getDomainName() As String

Return System.Environment.UserDomainName

End Function

‘‘‘ <summary>

''' Gets the currently logged in user's managers name from Active Directory.

''' </summary>

''' <returns>logged in user's managers name</returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function getManagerName() As String

Try

Dim UserName As String = System.Environment.UserName

Dim dSearcher As New DirectorySearcher("(mailNickname=" + UserName + ")")

Dim result As SearchResult = dSearcher.FindOne

If result Is Nothing Then

Return "Error finding user " + UserName + " using Active Directory."

End If

Dim employeeEntry As DirectoryEntry = result.GetDirectoryEntry

'\*\* find manager's LDAP DN from the DirectoryEntry

Dim strManagerDN$ = employeeEntry.Properties("manager").Value.ToString()

'\*\* use the DN to create a new DirectoryEntry

Dim managerEntry As New DirectoryEntry("LDAP://" + strManagerDN$)

'\*\* get the alias of the manager

Dim strManagerAlias$ = managerEntry.Properties("mailNickname").Value.ToString()

Return strManagerAlias

Catch ex As Exception

Return "ERROR: " + ex.Message

End Try

End Function

‘‘‘ <summary>

''' Gets the current thesaurus' name

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <returns>thesaurus name</returns>

''' <remarks></remarks>

#### <WebMethod()> Public Function getCurrentThesaurus(ByVal SessionGuid$) As String

Return GV.getThesaurus(SessionGuid$)

End Function

‘‘‘ <summary>

''' Sets the thesaurus' to be used by ECM.

''' </summary>

''' <param name="SessionGuid">Tracking Session ID</param>

''' <param name="ThesaurusName">A valid thesaurus name.</param>

''' <remarks></remarks>

#### <WebMethod()> Public Sub setCurrentThesaurus(ByVal SessionGuid$, ByVal ThesaurusName As String)

GV.setThesaurus(SessionGuid, ThesaurusName$)

End Sub

Index:

B

BIncludeLibraryFilesInSearch, 36

C

CbCCaddr, 34

CbDateSelection, 35

CbDirs, 36

CbEvalCreateTime, 38

CbEvalWriteTime, 38

CbFilename, 36

CbFileTypes, 36

CbFolderFilter, 34

CbFromAddr, 33

CbFromName, 33

CbLibrary, 32

CbMeta1, 36

CbMeta2, 37

CbMeta3, 37

CbMeta4, 37

CbToAddr, 34

CbToName, 34

CDateSelection, 34

CkBusiness, 33

CkCreationDate, 35

CkDays, 33

CkIncludeAttachments, 33

CkLimitToExisting, 30

CkLimitToLib, 32

CkReceivedTime, 35

CkSearchMasterDocsOnly, 36

CkSentDate, 34

CkWeighted, 33

D

DtCreateDateEnd, 38

DtCreateDateStart, 38

DtLastWriteEnd, 38

DtLastWriteStart, 38

DtMailDateEnd, 35

DtMailDateStart, 35

G

gConn, 32

gConnStr, 30

gCurrUserGuidID, 30

genDocSearchSql, 7

genEmailSearchSql, 8

gErrMsg, 32

getConnectionString, 7, 10

getEmailImage, 12

getIsAdmin, 14

getLicenseStatus, 5

GetMachineIPAddr, 13

getSearchString, 13

getSourceImage, 12

getSqlDS, 11

getSqlDsXML, 11

gGurrUserLoginID, 30

gIsGlobalSearcher, 31

gSearchString, 30

gSql, 32

gThesauri, 32

gThesaurus, 31

gThesaurusWords, 31

I

iCountRecordsInTable, 12

IsAdmin, 30

isGlobalSearcher, 32

L

LibraryName, 36

M

MasterContentOnly, 32, 35

MinRating, 32, 35

MyContentOnly, 32, 35

N

NbrOfErrors, 32

R

RbMyEmails, 34

restoreDocument, 6

restoreEmail, 6

S

SearchDocuments, 29

SearchDocumentsXML, 30

SearchEmail, 8

SearchEmailXML, 9

setbDateSelection, 22

setBIncludeLibraryFilesInSearc, 25

setbrOfErrors, 16

setCbCCaddr, 21

setCbDateSelection, 23

setCbDirs, 25

setCbEvalCreateTime, 28

setCbEvalWriteTime, 29

setCbFilename, 25

setCbFileTypes, 25

setCbFolderFilter, 21

setCbFromAddr, 20

setCbFromName, 20

setCbLibrary, 17

setCbToAddr, 20

setCbToName, 21

setCkBusiness, 18

setCkCreationDate, 23

setCkDays, 19

setckIncludeAttachments, 19

setCkLimitToExisting, 19

setCkLimitToLib, 17

setCkReceivedTime, 24

setCkSearchMasterDocsOnly, 25

setCkSentDate, 22

setCkWeighted, 18

setClipBoardData, 17

setConnStr, 9

setCurrUserGuidID, 14

setDateLastAccessEnd, 28

setDateLastAccessStart, 28

setDateLastWriteEnd, 28

setDateLastWriteStart, 28

setDateMailDateEnd, 23

setDateMailDateStart, 24

setErrMsg, 16

setGurrUserLoginID, 29

setIncludeAttachments, 13

setIsAdmin, 14

setIsGlobalSearcher, 14

setLibraryName, 24

setMasterContentOnly, 14

setMetaDataSearchKey, 26, 27

setMetaDataSearchValue, 26, 27

setMinRating, 17

setMyContentOnly, 15

setRbMyEmails, 21

setSearchString, 13

setsGlobalSearcher, 16

setSql, 16

setTxtCCPhrase, 22

setTxtDays, 20

setTxtSearch, 18

setTxtSubject, 18

T

TargetThesarus, 31

ThesarusWords, 31

TxtCCPhrase, 34

TxtDays, 33

TxtMetaSearch1, 37

TxtMetaSearch2, 37

TxtMetaSearch3, 37

TxtSubject, 33

W

WriteErr, 5