GFI Product Guide

Archiver

Web Services Reference Document

GFI Software™

The information and content in this document is provided for informational purposes only and is provided "as is" with no warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. GFI Software is not liable for any damages, including any consequential damages, of any kind that may result from the use of this document. The information is obtained from publicly available sources. Though reasonable effort has been made to ensure the accuracy of the data provided, GFI makes no claim, promise or guarantee about the completeness, accuracy, recency or adequacy of information and is not responsible for misprints, out-of-date information, or errors. GFI makes no warranty, express or implied, and assumes no legal liability or responsibility for the accuracy or completeness of any information contained in this document.

If you believe there are any factual errors in this document, please contact us and we will review your concerns as soon as practical.

Document Version: V. 2.3.0

Last updated (month/day/year): 17/11/2014

1 Introduction

GFI Archiver provides four web services to clients. These web services can be used to create tools that retrieve data from GFI Archiver.

The four available web services are:

Service	Description
Outlook Web Service	This web service can be used to perform similar functionality as the Outlook Connector. To consume the web service one must enter the URL of GFI Archiver followed by: `services\outlook.asmx`.
Send Web Service	This web service can be used to perform email restore operations. To consume the web service one must enter the URL of GFI Archiver followed by: `services\send.asmx`.
Search Web Service	This web service can be used to perform searches on the data within GFI Archiver. To consume the web service one must enter the URL of GFI Archiver followed by: `services\search.asmx`.
Store Web Service	This web service can be used to retrieve information about the archive stores within GFI Archiver. To consume the web service one must enter the URL of GFI Archiver followed by: `services\store.asmx`.

To use these web services through Microsoft Visual Studio, add a web reference to the solution with the URLs specified above, depending on which web service is required.

Once the web reference is added, the methods available in each referenced web service can be accessed. On the next page a snippet of how to use the Outlook web service can be found.

1.1 Code snippet example: Using Outlook web service

```
CookieContainer cookie = new CookieContainer();

Outlook outlookWS = new Outlook();
outlookWS.Url = "http://127.0.0.1/archiver/services/outlook.asmx";
outlookWS.Credentials = System.Net.CredentialCache.DefaultCredentials;
outlookWS.CookieContainer = cookie;

User user = outlookWS.Connect();
Console.WriteLine("Connected to Outlook WebService @ {0} with user {1}",
outlookWS.Url, user.DisplayName);

Console.WriteLine("Get list of MailBoxes to which I have access:");
User[] users = outlookWS.GetAllowedMailBoxes();

foreach (User allowedUser in users)
{
  Console.WriteLine("\n user name: {0}; login: {1}; user id: {2}",
  allowedUser.DisplayName, allowedUser.Name, allowedUser.UserId);
}
```

In the code snippet above, the Outlook web service has been referenced and the methods can be accessed. In this example, the URL and credentials are set to use windows authentication. Network credentials can be used by substituting the line:

```
outlookWS.Credentials = System.Net.CredentialCache.DefaultCredentials;
```

with the following:

```
outlookWS.Credentials = new NetworkCredential("user", "password",
"domain");
```

When this change is made, the connection is tested and a user object is returned. The URL and display name of the user are displayed in the console window.

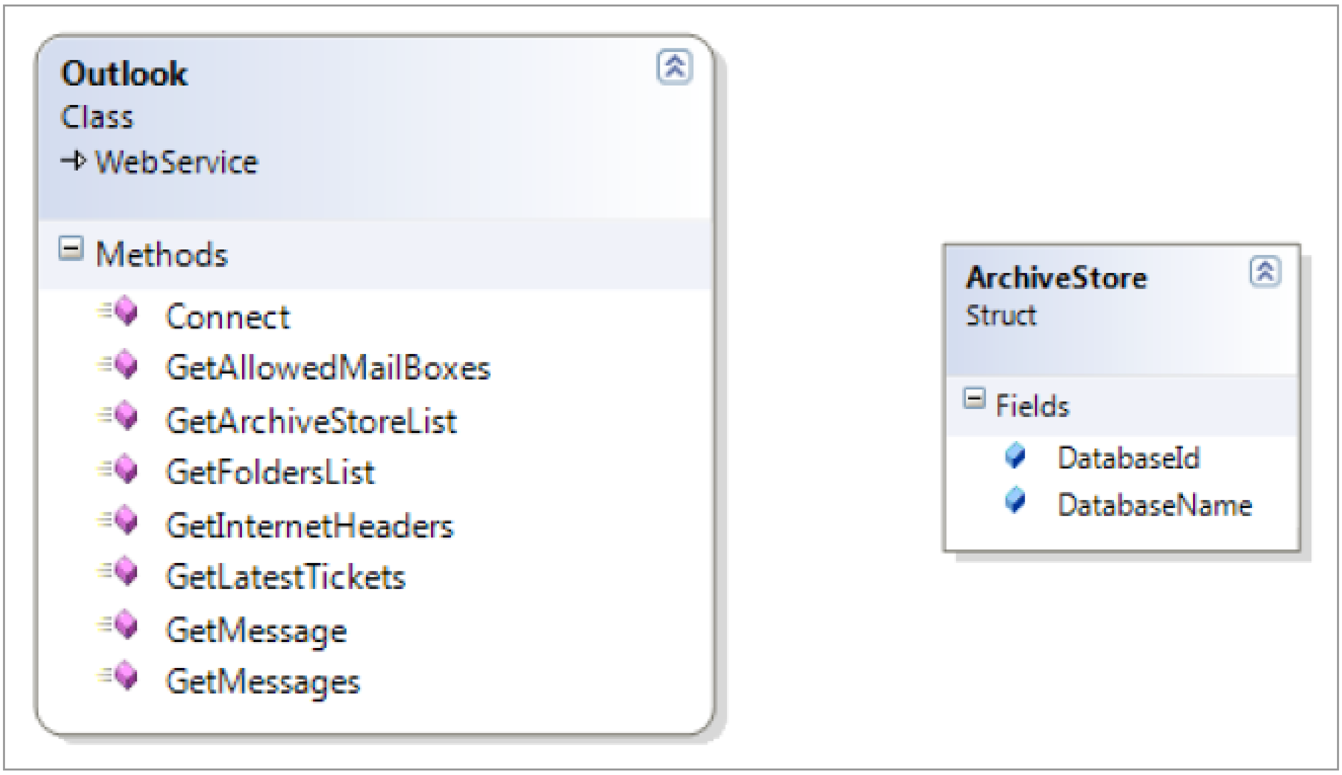
Once the user has been retrieved, the method `GetAllowedMailBoxes()` is called. This method returns a list of all the users that the current user has access too. Once the array of users is returned, a loop is performed to display all the users that the current user has access too.

To summarize, this snippet connects to the Outlook web service, confirms the connection and gets all the mailboxes that the current user has access too.

2 Outlook Service

The below section describes the Outlook web service. Each method is explained to get a better understanding of the service. To consume the web service one must enter the URL of GFI Archiver followed by: `services\outlook.asmx`.

Class Diagrams



Methods

Name	Description
Connect	Test connection and return current user details
GetAllowedMailBoxes	Get a list of mailboxes for all users to whom the current user has access
GetArchiveStoreList	Get a list of available database connections
GetFolderList	Get a list of folders for the given user
GetInternetHeaders	Get internet headers for the given message
GetLatestTickets	Used to return the latest timestamp for each database for the given user
GetMessage	Get Detailed Message, including body
GetMessages	Get cache updates

2.1 Connect

```
public MarUIHelper.User Connect()
```

Description:

This method tests the connection with the current user details retrieved within the method. It gathers and returns a user with the details explained below.

Parameters: *None*

Return Value: User

2.2 GetAllowedMailBoxes

```
public MarUIHelper.User[] GetAllowedMailBoxes()
```

Description:

This method gets all available mailboxes that the user has access to. For example, a user that is a member of the full access group will have access to all available mailboxes. If the user is set as a manager of a specific group, then the mailboxes that the manager has access to will be returned. In many instances, a user will only have access to his own mailbox and only that mailbox will be returned when this method is called.

Parameters: *None*

Return Value: User[]

2.3 GetArchiveStoreList

```
public ArchiveStore[] GetArchiveStoreList()
```

Description:

This method gets all the available archive stores that are visible. If an archive store does not "**allow users to browse and search this archive store**" then it will not be added to the array of archive stores.

Parameters: *None*

Return Value: ArchiveStores[]

2.4 GetFoldersList

```
public MarUIHelper.Folder[] GetFoldersList(string userId)
```

Description:

This method returns an array of folders for the user guid supplied. The main purpose of this method is to create the folder list in the tree panel. This means that this method will return all the folders of the supplied user.

Parameters:

string userId - Guid of user

Return Value: Folder[]

2.5 GetInternetHeaders

```
public string GetInternetHeaders(string connectionId, int messageId)
```

Description:

This method returns a string with the internet header of the message supplied from the specified database. To summarize, this method returns a string containing the internet header for the given `connectionid` and `messageid`.

Parameters:

- string connectionId - Database Guid
- int messageId - Marc specific message id for required email

Return Value: string internet header

2.6 GetLatestTickets

```
public MarUIHelper.Ticket[] GetLatestTickets(string userId)
```

Description:

This method is used to return an array of Ticket that will contain all databases and their associated latest timestamp for the given user. This method is also used to retrieve the latest updates on messages. This can be done by comparing the timestamp returned in the Ticket with the timestamp cached on the client side.

Parameters: string userId - Guid of user

Return Value: Ticket[]

2.7 GetMessage

```
public MarUIHelper.WSDetailedMessage GetMessage(string connectionId, int
messageId)
```

Description:

This method is used to return a detailed message from the database supplied. The message contains all the details regarding the message retrieved from the supplied `messageId`. This includes the body of the message and if the body is not found, it is returned as null.

Parameters:

- string connectionId - Database guid
- int messageId - Marc specific message id for required email

Return Value: WSDetailedMessage

2.8 GetMessages

```
public MarUIHelper.WSResultSet GetMessages(MarUIHelper.Ticket ticket,
DateTime startDate, DateTime endDate, int maxCount)
```

Description:

This method is used for normal synchronization and bulk synchronization of the client's cache. It gets a number of updates such as new messages, changes in tags and deleted messages.

When this method is called with the `Ticket`, `startDate`, `endDate`, and `maxCount` parameters, a ResultSet object is returned consisting of four elements: **Ticket**, **New messages**, **Deleted messages** and **Tag updates**.

The **Ticket** is used to keep track of what is already downloaded by keeping the latest timestamp, the **New messages** contain information about new messages added, the **Deleted messages** contain information about the messages deleted and **Tag updates** contain information about a change, for example, adding a label to a message.

Parameters:

Ticket ticket – Database guid

DateTime startDate – Start date from when emails should be retrieved

DateTime endDate – End date

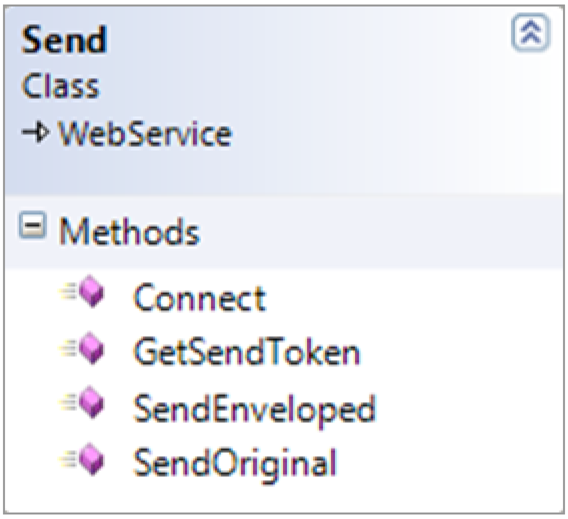
int maxCount - Maximum amount of update notifications ResultSet objects to be returned, this can be lower than the requested amount

Return Value: WSResultSet

3 Send Web Service

The following section describes the **Send Web Service**. Each method is explained to get a better understanding of the service. To consume the web service, enter the URL of GFI Archiver followed by: `services\send.asmx`.

Class Diagrams



Send
Class
→ WebService

Methods
- Connect
- GetSendToken
- SendEnveloped
- SendOriginal

Methods

Name	Description
Connect	Test connection and return string containing build number and version
GetSendToken	Generates a token for a continuous send operation
SendEnveloped	Restore email to mailbox in envelope (original as attachment)
SendOriginal	Restore email to mailbox in original format

3.1 Connect

```
public string Connect()
```

Description:

This method tests the connection by retrieving the product version and product build number.

Parameters: None

Return Value: string

3.2 GetSendToken

```
public Guid GetSendToken()
```

Description:

This method returns a guid that groups 'send operations' together. The guid will be used for a group of send operations. This token can be found in the cache. This method is required when performing a send operation.

Parameters: None

Return Value: Guid

3.3 SendEnveloped

```
public void SendEnveloped(string connectionGuid, int messageId, string
recipientEmail, string key, Guid seqGuid)
```

Description:

This method is used to send an archived email to the user’s mailbox as an attachment. It requires the database guid, the message id, the recipient’s email, the key which contains the session sequence number and the session sequence number. The key is used to verify that the GetSendToken was called and a token is available in the cache.

Parameters:

- string connectionGuid - database guid
- int messageId - MArc specific message id for required email
- string recipientEmail - email address of recipient
- string key - contains sequence guid
- Guid seqGuid - auditing session sequence number

Return Value: *None*

3.4 SendOriginal

```
public void SendOriginal(string connectionGuid, int messageId, string
recipientEmail, string key, Guid seqGuid)
```

Description:

This method is used to send an archived email to the user’s mailbox as the original message. It requires the database guid, the message id, the recipient’s email, the key which contains the session sequence number and the session sequence number. The key is used to verify that the GetSendToken was called and a token is available in the cache.

Parameters:

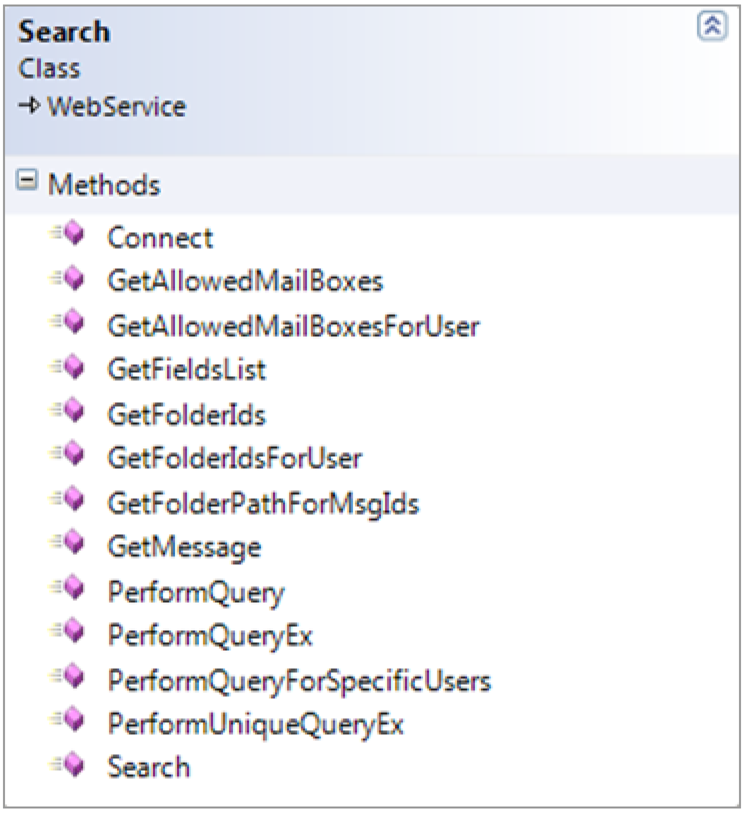
- string connectionGuid - database guid
- int messageId - MArc specific message id for required email
- string recipientEmail - email address of recipient
- string key - contains sequence guid
- Guid seqGuid - auditing session sequence number

Return Value: *None*

4 Search Web Service

The following section describes the **Search Web Service**. Each method is explained to get a better understanding of the service. To consume the web service one must enter the URL of GFI Archiver followed by: `services\search.asmx`.

Class Diagrams



Search
Class
→ WebService

⊟ Methods

- Connect
- GetAllowedMailBoxes
- GetAllowedMailBoxesForUser
- GetFieldsList
- GetFolderIds
- GetFolderIdsForUser
- GetFolderPathForMsgIds
- GetMessage
- PerformQuery
- PerformQueryEx
- PerformQueryForSpecificUsers
- PerformUniqueQueryEx
- Search

Name	Description
Connect	Test connection and return string containing build number and version
GetAllowedMailBoxes	Get a list of mailboxes for all users to whom the current user has access
GetAllowedMailBoxesForUser	Get a list of mailboxes for all users to whom the current user has access filtering with string provided
GetFolderList	Get a list of supported fields (stub)
GetInternetHeaders	Get internet headers for the given message
GetLatestTickets	Used to return the latest timestamp for each database for the given user
GetFolderIds	Get folder ids for user calling the method
GetFolderIdsForUser	Get folder ids for a specific user
GetFolderPathForMsgIds	Gets the folder paths for Message Ids specified in MessageInfo class array
GetMessage	Get archived message headers
PerformQuery	Perform search query
PerformQueryEx	Perform extended search query
PerformQueryForSpecificUsers	Perform search query for specific list of userIds
PerformUniqueQueryEx	Perform extended search query with Unique ID

4.1 Connect

```
public string Connect()
```

Description:

This method tests the connection by retrieving the `ProductVersion` and `ProductBuild`.

Parameters: *None*

Return Value: string

4.2 GetAllowedMailBoxes

```
public MarUIHelper.User[] GetAllowedMailBoxes()
```

Description:

This method gets all the available mailboxes that the user has access to. For example, a user within the full access group can access all available mailboxes and if a user is set as manager of a specific group, then the mailboxes that the manager has access to are returned. In many situations a user will only have access to his own mailbox and only that mailbox is returned when this method is called.

Parameters: *None*

Return Value: User[]

4.3 GetAllowedMailBoxesForUser

```
public MarUIHelper.User[] GetAllowedMailBoxesForUser(string strUserSearch)
```

Description:

This method gets all the mailboxes for the specified user. Only mailboxes to which the user has access are returned. For example, if the word “John” is specified, all mailboxes that match “John” are returned if the current user has access to those mailboxes. This method is very similar to **GetAllowedMailBoxes** but instead it filters the results depending on the string passed as a parameter.

Parameters: string strUserSearch

Return Value: User[]

4.4 GetFieldsList

```
public IndexFields GetFieldsList()
```

Description:

This method returns a new instance of **IndexFields**. This is a list of supported fields such as attachment, body, owners and priority.

Parameters: *None*

Return Value: IndexFields

4.5 GetFolderIds

```
public string[] GetFolderIds(FolderOpt[] folderOptions)
```

Description:

This method is used to retrieve the IDs of the folders for the user calling the method. The method calls `GetFolderIdsForUser` passing two parameters, `FolderOpt[]` with the folder options which consists of the folder path and whether to search for subfolders and a string with the user guid. `GetFolderIdsForUser` will return an array of strings containing the folder ids.

Parameters: FolderOpt[] folderOptions

Return Value: string[]

4.6 GetFolderIdsForUser

```
public string[] GetFolderIdsForUser(FolderOpt[] folderOptions, string
userGuid)
```

Description:

This method is used to retrieve an array of strings containing the folder IDs of the folders for the required user. The folderOptions parameter contains the path of the folder and whether to check for subfolders.

Parameters: FolderOpt[] folderOptions

Return Value: string[]

4.7 GetFolderPathForMsgIds

```
public MarUIHelper.FolderPath[]

GetFolderPathForMsgIds(MarUIHelper.MessageInfo[] messagesInfo)
```

Description:

This method gets the folder paths for the message IDs supplied in the `MessageInfo` class array.

Also this method is used by the Bulk export tool search so that the `FolderPath` for particular messages can be obtained. The method is called with a `MessageInfo`[] class array with a list of messages containing the guid of the database, user and the message id. The folder path is then returned for each of these messages.

Parameters: MessageInfo[] messageInfo

Return Value: FolderPath[]

4.8 GetMessage

```
public Message GetMessage(string connectionId, int messageId, out string
messageUrl, out string sourceUrl)
```

Description:

This method gets the archived message headers. Using the `connectionId` (which is the database guid) and the `messageId` (which is the MArc specific messageId for the required email) the information for the header can be retrieved. This information is then returned in a Message object. This message object contains information about the message such as size, owner, sent date, archived date and so on.

Parameters:

- string connectionId - Database guid
- int messageId - MArc specific message id for required email

- out string messageUrl - contains link to mailview of email
- out string sourceUrl - contains link to source of email

Return Value: Message

4.9 PerformQuery

```
public SearchEmailResults PerformQuery(SearchOptions parameters, out Guid
seqNo)
```

Description:

This method performs a search query. When this method is called it is passed a `SearchOptions` object that contains all the information about the search. A `SearchEmailResults` object is returned with all the information regarding the results of the search performed with the `SearchOptions` object.

Parameters:

- SearchOptions parameters - Contains information about search done by user
- out Guid seqNo - auditing session sequence number

Return Value: SearchEmailResults

4.10 PerformQueryEx

```
public SearchEmailResults PerformQueryEx(SearchOptions parameters, string
rawQuery, string xFilter, int limit, List<string> ownerids, out Guid
seqNo)
```

Description:

This method performs an extended search query. It works the same as `PerformQuery` but instead it accepts raw search queries. These raw queries also accept filtering and limit the number of results returned.

Parameters:

- SearchOptions parameters - Contains information about search done by user
- string rawQuery - a raw search query
- string xFilter - filter raw search query
- int limit - number of emails
- List<string> ownerids - list of guids of owners
- out Guid seqNo - auditing session sequence number

Return Value: SearchEmailResults

4.11 PerformQueryForSpecificUsers

```
public SearchEmailResults PerformQueryForSpecificUsers(SearchOptions
parameters, List<string> users, out Guid seqNo)
```

Description:

This method performs a search query for specific users. When this method is called it is passed a `SearchOptions` object that contains all the information about the search. A `SearchEmailResults` object is returned with all the information regarding the results of the search performed with the `SearchOptions` object.

Parameters:

- SearchOptions parameters - Conatins information about search done by user
- List<string> users - list of guids of users
- out Guid seqNo - auditing session sequence number

Return Value: SearchEmailResults

4.12 PerformUniqueQueryEx

```
public SearchEmailResults PerformUniqueQueryEx(SearchOptions parameters,
List<string> ownerids, UniqueMessage uniqueMsg, out Guid seqNo)
```

Description:

This method performs an extended search query with unique ID, it works similar to `PerformQuery` but instead it accepts a `UniqueMessage` object. This object contains the database guid and the message ID. This is the identification code of the email.

Parameters:

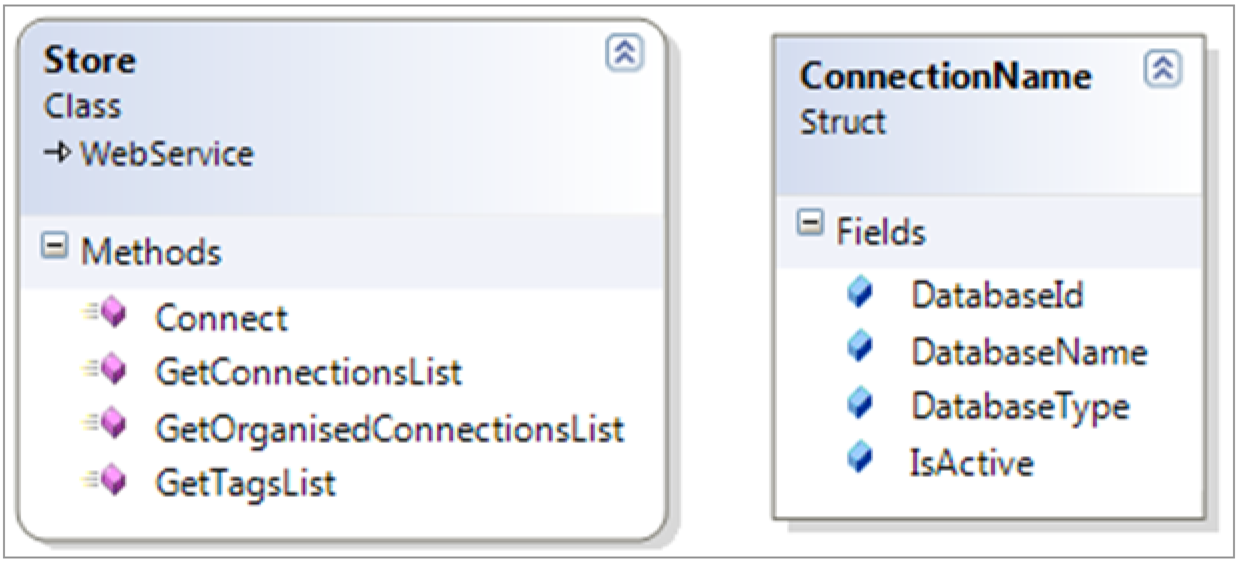
- SearchOptions parameters - Contains information about search done by user
- List<string> ownerids - list of guids of owners
- UniqueMessage uniqueMsg - contains database guid, message id and message type
- out Guid seqNo - auditing session sequence number

Return Value: SearchEmailResults

5 Store Web Service

The following section describes the **Store Web Service**. Each method is explained to get a better understanding of the service. To consume the web service enter the URL of GFI Archiver followed by: `services\store.asmx`.

Class Diagrams



Methods

Name	Description
Connect	Test connection and return string containing build number and version
GetConnectionsList	Get a list of available database connections
GetOrganisedConnectionsList	Get a list of available database connections organized in Active first and last entered afterwards
GetTagsList	Get a list of the current user's and retention tags

5.1 Connect

```
public string Connect()
```

Description:

This method tests the connection by retrieving the product version and product build number.

Parameters: *None*

Return Value: string

5.2 GetConnectionsList

```
public ConnectionName[] GetConnectionsList()
```

Description:

This method gets a list of all the databases. It returns an array of `ConnectionName` that contains, Database name, Database id, Database type and whether it is active or not.

Parameters: *None*

Return Value: ConnectionName[]

5.3 GetOrganisedConnectionsList

```
public ConnectionName[] GetOrganisedConnectionsList()
```

Description:

This method gets a list of all the databases organized in active first and last entered afterward. It returns an array of `ConnectionName` that contains, Database name, Database id, Database type and whether it is active or not.

Parameters: *None*

Return Value: ConnectionName[]

5.4 GetTagsList

```
public global::Store.Dal.Elements.Tag[] GetTagsList()
```

Description:

This method gets a list of all the retention tags set. It returns an array of a Tag object which contains the name, type and condition of each retention policy set.

Parameters: None

Return Value: Tag[]

6 Outlook Web Service Classes

Class	Description
User	This class contains information about a user. It contains the following properties: » string DisplayName – Active directory display name » string UserId – Guid of user » string Name – Active directory username
Archive Store	This class contains information about an archive store. It contains the following properties: » string DatabaseName – name of database » string DatabaseId – guid of database
Folder	This class contains information about a folder. Folders are stored as labels in Marc database terms, so that structure can be preserved. It contains the following properties: » string FolderId - Guid » string FolderName - Guid
Ticket	This class contains information about a ticket. A ticket contains the following properties: » long Timestamp – Latest timestamp » long CeilingTimeStamp – timestamp that’s set to Int64.MinValue when no more data is returned » string UserId – guid of user » Guid DatabaseId – guid of database

Class	Description
WSDetailedMessage	This class contains a number of properties that contain information about a message. It contains data such as the message id, archive date, Size and so on. The properties are described below: » WSAddress[] AddressCollection - This property holds an array of WSAddress. » DateTime? ArchivedDate - This property contains the archived date of the message » Attachment[] AttachmentCollection - This property holds and array of Attachment » WSBody Body - This property holds a WSBody object. » WSMessageDirection Direction - This property holds an enum of WSMessageDirection. » bool Encrypted - bool describing whether the message is encrypted or not » string InetMessageId - Contains a string with the mime internet message id » int MessageId - Integer containing the message id » int Priority - Integer containing priority of message » DateTime ReceivedDate - Contains the received date of the message » DateTime SentDate- Contains the sent date of the message » bool Signed - Contains a bool whether the message is signed or not » int Size - Integer with the size of the email » string Subject - string containing the subject of message » WSTag[] TagCollection - This property holds an array of WSTag » string ThreadIndex - string containing the thread index of the message » string ThreadTopic - string containing the thread topic of the message » long TimeStamp - contains timestamp of message
WSAddress	This class contains information about a user's address and consists of three properties: » string DisplayName – contains display name of user » string Email – contains email address of user » WSAddressType Type – enum of WSAdessType
WSAddressType	This is an enum that contains the following values: » From » To » CC » EFrom » ETo » EExportOwner » UnknownAddress » Sender

Class	Description
WSMessageDirection	This is an enum that contains the following values: » Imported » Internal » Inbound » Outbound » Mixed » Unknown
Attachment	This class contains information the attachment and consists of five properties: » int AttachmentId - contains attachment id » string ContentId - contains content id value of the attachment » string ContentLocation - contains attachment content location » string FileName - name of attachment » int Size - size of attachment
WSBody	This class contains information about the body of the message and consists of three properties: » byte[] BodyData - contains the message » string CharSet - contains character encoding » BodyType Type - enum of BodyType.
BodyType	This is an enum that contains the following values: » Txt » Html » Empty
WSTag	This class contains information about the tags set on the message and consists of four properties: » string FolderId - contains the id of the folder » string Name - contains the name of the tag » WSTagType Type - enum of WSTagType » string Value - contains a string of the label set
WSTagType	This is an enum that contains the following values: » SystemTag » CustomTag » UserTag » None

Class	Description
WSResultSet	This class contains a number of properties that contain information about a Result set. It contains data such as Deleted messages, new messages, tag updates and ticket. The properties are described below: » WSDeletedMessage[] DeletedMessages - This property holds an array of WSDeletedMessage. » WSSimpleMessage[] NewMessages - This property holds an array of WSSimpleMessage. » WSTagUpdate[] TagUpdates - This property holds an array of WSTagUpdate » Ticket Ticket - This property holds a Ticket object
WSDeletedMessage	This class contains information about a deleted message and consists of two properties: » int MessageId - contains the id of the message » long TimeStamp - contains the timestamp of the message
WSSimpleMessage	This class contains information about a message and consists of sixteen properties: » DateTime? ArchivedDate - contains archive date of email » bool HasAttachments - bool whether a message has an attachment » int MessageId - int containing message id » int Priority - int with priority of email » DateTime ReceivedDate - contains received date of email » string Recipients - contains a string of recipients » string Sender - contains a string with the sender » string SenderName - contains a string the name of the sender » DateTime SentDate - contains the sent date of the message » int Size - size of the message » string Subject - subject of message » WSTag[] TagCollection - This property holds an array of WSTag » string ThreadIndex - string containing the thread index of the message » string ThreadTopic - string containing the thread topic of the message » long TimeStamp - contains timestamp of message
WSTagUpdate	This class contains information about the updates of tags and contains six properties: » string FolderId - contains guid of folder » int MessageId - contains message id » string TagName - contains the name of the tag » WSTagType TagType - Enum of WSTagType » string TagValue - contains string with label set » long TimeStamp - timestamp of label

7 Search Web Service Classes

Class	Description

IndexFields	This is a struct that contains information about the index fields. It contains the following constants:
	» const string ATTACHMENT = "attachment";
	» const string ATTACHMENTANDBODY = "attachmentandbody";
	» const string ATTACHMENTS_TEXT = "attachmentstext";
	» const string BCC = "bcc";
	» const string BODY = "body";
	» const string DBID = "marcdatabaseid";
	» const string HAS_ATTACHMENTS = "hasattachments";
	» const string MARCID = "marcid";
	» const int MAX_DISPLAYNAME_LEN = 50;
	» const int MAX_FIELDVAL_LEN = 250;
	» const string MSGDATE_FORMAT = "0000000000";
	» const string MSGID = "msgid";
	» const string MSGSIZE = "msgsize";
	» const string MSGSIZE_FORMAT = "0000000000";
	» const string MSGSIZE_KB = "msgsizekb";
	» const string OIDS = "oids";
	» const string OWNERS = "owners";
	» const string PRIMARYOID = "prioid";
	» const string PRIORITY = "priority";
	» const string RECVDATE = "recvdate";
	» const string RECVDATE_DAYS = "recvdated";
	» const string RECVDATE_SECONDS = "recvdates";
	» const string SENDER = "sender";
	» const string SENTDATE = "sentdate";
	» const string SENTDATE_DAYS = "sentdated";
	» const string SENTDATE_SECONDS = "sentdates";
	» const string SUBJECT = "subject";
	» const string SUBJECT_DISPLAY = "subjectd";
	» const string THREAD_ID = "threadid";
	» const string TO = "to";
	» const string XML_SENDER = "xmlsender";
	» const string XML_TO = "xmlto";

Class	Description
FolderOpt	This class contains information about the options of a folder and has the following properties: » bool boolSearchSubFolders – whether to search subfolders or not » string strFolderPath – path of the folder
FolderPath	This class contains information about the folder. It contains the following properties: » string Path – contains path » string Type – contains type of folder
MessageInfo	This class contains information about a message. It contains the following properties: » guid Connectionid – Guid of database » int messageid – Marc specific message id for required email » string userid – user guid
Message	This class contains detailed information about a message. It contains the following properties: » Address[] AddressCollection – Contains an array of Address. » DateTime ArchivedDate – archived date of message » Attachment[] AttachmentCollection – Contains an array of Attachment » string DuplicateThreadIndex – string with duplicate thread index » bool Encrypted – whether message is encrypted or not » string InetMessageId - Contains a string with the mime internet message id » int MessageId – int with message id » Owner[] OwnersCollection – Contains an array of Owner » int ParentId – int of parent id » int Priority – int containing priority » DateTime ReceivedDate – received date of message » string RuleId – id of rule applied to message » DateTime SentDate – sent date of message » bool Signed – bool whether message is signed or not » int Size – size of message » string Subject – subject of message » int ThreadId – int thread id » long TimeStamp – timestamp of message
Address	This class contains information about a user's address and consists of three properties: » string DisplayName – contains display name of user » string Email – contains email address of user » AddressType Type – enum of AddessType

Class	Description
AddressType	This is an enum that contains the following values: » From » To » CC » EFrom » ETo » EExportOwner » UnknownAddress » Sender
Attachment	This class contains information the attachment and consists of five properties: » int AttachmentId – contains attachment id » string ContentId – contains content id value of the attachment » string ContentLocation – contains attachment content location » string FileName – name of attachment » int Size - size of attachment
Owner	This class contains information about the owners of the message and contains the following properties: » bool IsPrimary - bool whether owner is primary » string Name – name of owner » AddressType Type - enum of AddessType » string UserName – username of owner
SearchEmailResults	This class contains information about the result of a search. It contains the following properties: » bool Done – bool whether search is done or not » long Hitcount – the amount of hits » List<SimpleMessage> Hits - list of SimpleMessage » string Q – query string » bool SearchInAttachmentSupported – bool whether searching in attachments is supported » bool ThresholdReached – bool whether the threshold is reached

Class	Description
SimpleMessage	This class contains information about a message. It contains the following properties: » Guid DatabaseId - database guid » bool HasAttachments - whether email has an attachment or not » int MarcId - mailarchiver message id » string MsgID - message id of message » string Recipients - recipients of message » DateTime RecvDate - received date of message » string Sender - sender of message » DateTime SentDate - sent date of message » int Size - size of message » string Subject - subject of message
SearchOptions	This class contains information about the search options. It consists of the following properties: » bool Ascending - bool whether ascending or not » List<string> DBIds - list of database ids » List<FieldOpt> Fields - list of FieldOpt » int HitOffset - page offset » int HitsPerPage - number of hits per page » List<RangeFieldOpt> RangeFields - list of RangeFieldOpt » bool RestrictSearch - whether it is a restricted search or not » Search.Meta.SearchQuery SearchQuery - SearchQuery » bool Sort - whether to sort or not » string SortField - which field to sort
FieldOpt	This class is used for the field option set in the search. It consists of the following properties: » booleanflag_t boolop - enum booleanflag_t » string fieldname - fieldname set » string query - query set
booleanflag_t	This is an enum that contains the following values: » Null » AND » ANDNOT » OR

Class	Description
RangeFieldOpt	This class is used for the field options that have a range such as date and message size. It consists of the following properties: » booleanflag_t boolop - enum booleanflag_t » long end - end value » string fieldname - field name set » long start - start value
Search.Meta.SearchQuery	This class is used for searches within folders and labels and consists of the following properties: » string[] Folders - list of folders » string[] NotFolders - list of excluded folders » Tag[] NotTags - list of excluded tags » bool Or - bool whether the OR operator is used » Tag[] Tags - list of tags
Tag	This class contains information about the tags available. It consists of the following properties: » string Name - name of tag » TagType Type - enum of TagType » string Value - value of tag » int CompareTo(Tag other) - used to compare tags » bool Equals(Tag other) - used to check is tags are the same
TagType	This is an enum that contains the following values: » SystemTag » CustomTag » UserTag » None
UniqueMessage	This class contains information about a message. It contains the following properties: » string DBid - contains the database guid » int MessageId - contains message id » MessageType MessageType - enum of MessageType
MessageType	This is an enum that contains the following values: » EML » FAX

8 Store Web Service

Class	Description
ConnectionName	This struct is used for connection details. It contains the following properties: » string DatabaseName – Name of database » string DatabaseId – Guid of database » DatabaseType DatabaseType – enum of DatabaseType » bool IsActive – whether database is active or not
DatabaseType	This is an enum that contains the following values: » Access » MsSql » MySq » FS » None » FireBird » Marc3
Store.Dal.Elements.Tag	This class contains information about a tag. It consists of the following parameters: » string Name - Name of tag » TagType Type - enum of TagType » string Value - condition

USA, CANADA AND CENTRAL AND SOUTH AMERICA

4309 Emperor Blvd, Suite 400, Durham, NC 27703, USA
Telephone: +1 (888) 243-4329
Fax: +1 (919) 379-3402
ussales@gfi.com

UK AND REPUBLIC OF IRELAND

Magna House, 18-32 London Road, Staines-upon-Thames, Middlesex, TW18 4BP, UK
Telephone: +44 (0) 870 770 5370
Fax: +44 (0) 870 770 5377
sales@gfi.co.uk

EUROPE, MIDDLE EAST AND AFRICA

GFI House, Territorials Street, Mriehel, BKR 3000, Malta
Telephone: +356 2205 2000
Fax: +356 2138 2419
sales@gfi.com

AUSTRALIA AND NEW ZEALAND

83 King William Road, Unley 5061, South Australia
Telephone: +61 8 8273 3000
Fax: +61 8 8273 3099
sales@gfiap.com

GFI Software™