# How to Search for Items by Retention Period in Office 365 Exchange Online

## Introduction

Now we can easily search for email messages by retention policy in Outlook. But this feature is not available in Outlook Web App(OWA). In this application, we demonstrate how to search for email messages by retention policy enabled in Office 365 Exchange Online by using Exchange Web Service Managed API.

We use the following extend properties to search and get the information: 1. PidTagRetentionPeriod (Property ID:0x301A);2. PidTagRetentionDate (Property ID:0x301C).

## Running the Sample

Press F5 to run the sample, you will get the following result.

First, we input the retention periods that we want to search by. We can input multiple periods that are separated by space.



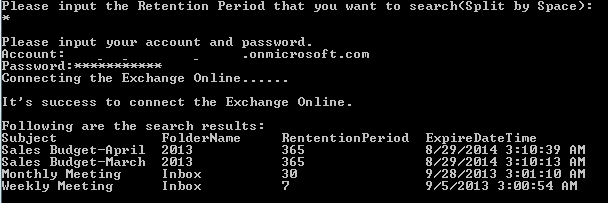
Then, we use our account to connect to the Exchange Online.



Once the connection is successful, we'll get the search results.



We can also input char-‘\*’ first to get all the messages that are applied the retention policies.



## Using the Code

1. Set two extend properties.

Use two extend properties to get the retention period and the expiring date.

|  |
| --- |
| -Code block start-  --C# code snippet start--  const Int32 pidTagRetentionPeriod = 12314;  ExtendedPropertyDefinition exPropRetentionPeriod =  new ExtendedPropertyDefinition(pidTagRetentionPeriod, MapiPropertyType.Integer);  const Int32 pidTagRetentionDate = 12316;  ExtendedPropertyDefinition exPropRetentionDate =  new ExtendedPropertyDefinition(pidTagRetentionDate, MapiPropertyType.SystemTime);  --C# code snippet end--  --VB code snippet start--  Const pidTagRetentionPeriod As Int32 = 12314  Dim exPropRetentionPeriod As New ExtendedPropertyDefinition(pidTagRetentionPeriod,  MapiPropertyType.Integer)  Const pidTagRetentionDate As Int32 = 12316  Dim exPropRetentionDate As New ExtendedPropertyDefinition(pidTagRetentionDate,  MapiPropertyType.SystemTime)  --VB code snippet end--  -Code block end- |

2. Set the search filters.

Set the search filters for each retention period.

|  |
| --- |
| -Code block start-  --C# code snippet start--  searchFilterCollection.LogicalOperator = LogicalOperator.Or;  foreach (Int32 period in periods)  {  SearchFilter.IsEqualTo searchFilter = new SearchFilter.IsEqualTo(exPropRetentionPeriod, period);  searchFilterCollection.Add(searchFilter);  }  --C# code snippet end--  --VB code snippet start--  searchFilterCollection.LogicalOperator = LogicalOperator.Or  For Each period As Int32 In periods  Dim searchFilter As New SearchFilter.IsEqualTo(exPropRetentionPeriod, period)  searchFilterCollection.Add(searchFilter)  Next period  --VB code snippet end--  -Code block end- |

3. Set search folder

Use search folder to search for messages.

|  |
| --- |
| -Code block start-  --C# code snippet start--  SearchFolder searchFolder = new SearchFolder(service);  searchFolder.SearchParameters.RootFolderIds.Add(rootFolder);  searchFolder.SearchParameters.Traversal = SearchFolderTraversal.Deep;  searchFolder.SearchParameters.SearchFilter = searchFilterCollection;  searchFolder.DisplayName = DateTime.Now.ToString("yyyyMMddhhmmss");  searchFolder.Save(WellKnownFolderName.SearchFolders);  --C# code snippet end--  --VB code snippet start--  Dim searchFolder As New SearchFolder(service)  searchFolder.SearchParameters.RootFolderIds.Add(rootFolder)  searchFolder.SearchParameters.Traversal = SearchFolderTraversal.Deep  searchFolder.SearchParameters.SearchFilter = searchFilterCollection  searchFolder.DisplayName = Date.Now.ToString("yyyyMMddhhmmss")  searchFolder.Save(WellKnownFolderName.SearchFolders)  --VB code snippet end--  -Code block end- |

4. Get search results

Get search results from search folder and display the information.

|  |
| --- |
| -Code block start-  --C# code snippet start--  do  {  findResults=searchFolder.FindItems(itemView);  foreach (Item findResult in findResults)  {  Object rPeriod = findResult.ExtendedProperties[0].Value;  Object expireDateTime = findResult.ExtendedProperties[1].Value;  if (!mailboxFolderNames.ContainsKey(findResult.ParentFolderId.UniqueId))  {  Folder folder = Folder.Bind(service, findResult.ParentFolderId);  mailboxFolderNames.Add(findResult.ParentFolderId.UniqueId, folder.DisplayName);  }  String folderName = mailboxFolderNames[findResult.ParentFolderId.UniqueId];  Console.WriteLine("{0,-20}{1,-15}{2,-18}{3}", findResult.Subject, folderName, rPeriod, expireDateTime);  }  } while (findResults.MoreAvailable);  --C# code snippet end--  --VB code snippet start--  Do  findResults = searchFolder.FindItems(itemView)  For Each findResult As Item In findResults  Dim rPeriod As Object = findResult.ExtendedProperties(0).Value  Dim expireDateTime As Object = findResult.ExtendedProperties(1).Value  If Not mailboxFolderNames.ContainsKey(findResult.ParentFolderId.UniqueId) Then  Dim folder As Folder = folder.Bind(service, findResult.ParentFolderId)  mailboxFolderNames.Add(findResult.ParentFolderId.UniqueId, folder.DisplayName)  End If  Dim folderName As String = mailboxFolderNames(findResult.ParentFolderId.UniqueId)  Console.WriteLine("{0,-20}{1,-15}{2,-18}{3}", findResult.Subject, folderName,  rPeriod, expireDateTime)  Next findResult  Loop While findResults.MoreAvailable  --VB code snippet end--  -Code block end- |

## More Information

[EWS Managed API 2.0](http://msdn.microsoft.com/en-us/library/dd633709(v=exchg.80).aspx)

[[MS-OXPROPS]: Exchange Server Protocols Master Property List](http://msdn.microsoft.com/en-us/library/cc433490(v=exchg.80).aspx)