# How to Add Delegates in Office 365 Exchange Online

## Introduction

Currently, you can easily add delegates in Outlook. But you will find this feature is not available in Outlook Web App (OWA). In this application, we will demonstrate how to add multi delegates in Office 365 Exchange Online.

1. Get the addresses of delegates;

2. Get the addresses of primary accounts;

3. Set the ImpersonatedUserId property if the login account has the impersonation permission.

4. Add all the delegates into all the primary accounts.

## Running the Sample

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

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



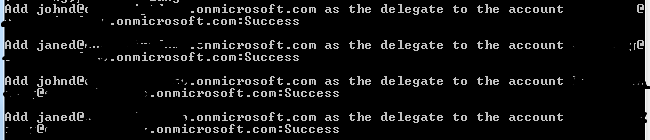
Then, we need the identities of the delegates.



And if we have the impersonation permission, we can add the delegates to multi accounts by setting the ImpersonatedUserId property.



The following is the result:



If we don’t have the impersonation permission, we can directly press Enter to add the delegates to the login account.

## Using the Code

Before we add the delegates, we need the addresses of the delegates.

|  |
| --- |
| -Code block start-  --C# code snippet start--  Console.WriteLine("Please input the user identity(s) that were used as the delegates:");  String delegateInfo = Console.ReadLine();  // We get the addresses related to the identities of delegates.  List<String> delegateIdentities = new List<string>();  if (!String.IsNullOrWhiteSpace(delegateInfo))  {  // You can input the "EXIT" to exit.  if (delegateInfo.ToUpper().CompareTo("EXIT") == 0)  {  return;  }  foreach (String delegateIdentity in delegateInfo.Split(','))  {  NameResolutionCollection nameResolutions =  service.ResolveName(delegateIdentity, ResolveNameSearchLocation.DirectoryOnly, true);  if (nameResolutions.Count != 1)  {  Console.WriteLine("{0} is invalid user identity as the delegate.",  delegateIdentity);  }  else  {  delegateIdentities.Add(nameResolutions[0].Mailbox.Address);  }  }  if (delegateIdentities.Count == 0)  {  Console.WriteLine("There's not any valid user identity as the delegate.");  continue;  }  }  else  {  Console.WriteLine("The delegates cannot be null!");  continue;  }  --C# code snippet end--  --VB code snippet start--  Console.WriteLine("Please input the user identity(s) that were used as the delegates:")  Dim delegateInfo As String = Console.ReadLine()  ' We get the addresses related to the identities of delegates.  Dim delegateIdentities As New List(Of String)()  If Not String.IsNullOrWhiteSpace(delegateInfo) Then  ' You can input the "EXIT" to exit.  If delegateInfo.ToUpper().CompareTo("EXIT") = 0 Then  Return  End If  For Each delegateIdentity As String In delegateInfo.Split(","c)  Dim nameResolutions As NameResolutionCollection =  service.ResolveName(delegateIdentity, ResolveNameSearchLocation.DirectoryOnly, True)  If nameResolutions.Count <> 1 Then  Console.WriteLine("{0} is invalid user identity as the delegate.",  delegateIdentity)  Else  delegateIdentities.Add(nameResolutions(0).Mailbox.Address)  End If  Next delegateIdentity  If delegateIdentities.Count = 0 Then  Console.WriteLine("There's not any valid user identity as the delegate.")  Continue Do  End If  Else  Console.WriteLine("The delegates cannot be null!")  Continue Do  End If  --VB code snippet end--  -Code block end- |

If we have impersonation permission, we can add the delegates to multi accounts by setting the ImpersonatedUserId property.

|  |
| --- |
| -Code block start-  --C# code snippet start--  Console.WriteLine("Please input the user identity you want to set the " +  "delegates(or you can directly press Enter to set delegates in current account):");  String primaryInfo = Console.ReadLine();  if (!String.IsNullOrWhiteSpace(primaryInfo))  {  // You can input the "EXIT" to exit.  if (primaryInfo.ToUpper().CompareTo("EXIT") == 0)  {  return;  }  String[] primaryIdentities = primaryInfo.Split(',');  foreach (String primaryIdentity in primaryIdentities)  {  // If the user identity is valid, we will set it as the ImpersonatedUserId.  NameResolutionCollection nameResolutions =  service.ResolveName(primaryIdentity, ResolveNameSearchLocation.DirectoryOnly, true);  if (nameResolutions.Count != 1)  {  Console.WriteLine("{0} is invalid user identity.", primaryIdentity);  }  else  {  String emailAddress = nameResolutions[0].Mailbox.Address;  service.ImpersonatedUserId =  new ImpersonatedUserId(ConnectingIdType.SmtpAddress, emailAddress);  foreach (String delegateIdentity in delegateIdentities)  {  AddAccountDelegates(service, emailAddress, delegateIdentity,  permissionLevelName, permissionLevel);  }  }  }  }  --C# code snippet end--  --VB code snippet start--  Console.WriteLine("Please input the user identity you want to set the " &  "delegates(or you can directly press Enter to set delegates in current account):")  Dim primaryInfo As String = Console.ReadLine()  If Not String.IsNullOrWhiteSpace(primaryInfo) Then  ' You can input the "EXIT" to exit.  If primaryInfo.ToUpper().CompareTo("EXIT") = 0 Then  Return  End If  Dim primaryIdentities() As String = primaryInfo.Split(","c)  For Each primaryIdentity As String In primaryIdentities  ' If the user identity is valid, we will set it as the ImpersonatedUserId.  Dim nameResolutions As NameResolutionCollection =  service.ResolveName(primaryIdentity, ResolveNameSearchLocation.DirectoryOnly,  True)  If nameResolutions.Count <> 1 Then  Console.WriteLine("{0} is invalid user identity.", primaryIdentity)  Else  Dim emailAddress As String = nameResolutions(0).Mailbox.Address  service.ImpersonatedUserId =  New ImpersonatedUserId(ConnectingIdType.SmtpAddress, emailAddress)  For Each delegateIdentity As String In delegateIdentities  AddAccountDelegates(service, emailAddress, delegateIdentity,  permissionLevelName, permissionLevel)  Next delegateIdentity  End If  Next primaryIdentity  --VB code snippet end--  -Code block end- |

If we don’t have the impersonation permission, we can directly press Enter to add the delegates to the login account.

|  |
| --- |
| -Code block start-  --C# code snippet start--  else  {  // We can also directly press Enter to add the delegates to the login account.  foreach (String delegateIdentity in delegateIdentities)  {  AddAccountDelegates(service, currentAddress, delegateIdentity,  permissionLevelName, permissionLevel);  }  }  --C# code snippet end--  --VB code snippet start--  Else  ' We can also directly press Enter to add the delegates to the login account.  For Each delegateIdentity As String In delegateIdentities  AddAccountDelegates(service, currentAddress, delegateIdentity,  permissionLevelName, permissionLevel)  Next delegateIdentity  End If  --VB code snippet end--  -Code block end- |

We add the delegate to the specified account.

|  |
| --- |
| -Code block start-  --C# code snippet start--  DelegateUser delegateUser = new DelegateUser(delegateAddress);  // Set the permission property of the delegate user.  foreach (PropertyInfo property in typeof(DelegatePermissions).GetProperties())  {  if (String.Compare(property.Name, permissionLevelName) == 0)  {  property.SetValue(delegateUser.Permissions, permissionLevel);  break;  }  }  Mailbox emailBox = new Mailbox(primaryAddress);  Collection<DelegateUserResponse> responses = service.AddDelegates(emailBox,  MeetingRequestsDeliveryScope.DelegatesAndSendInformationToMe, delegateUser);  foreach (DelegateUserResponse response in responses)  {  Console.WriteLine("Add {0} as the delegate to the account {1}:{2}",  delegateAddress, primaryAddress, response.Result);  Console.WriteLine();  }  --C# code snippet end--  --VB code snippet start--  Dim delegateUser As New DelegateUser(delegateAddress)  ' Set the permission property of the delegate user.  For Each [property] As PropertyInfo In GetType(DelegatePermissions).GetProperties()  If String.Compare([property].Name, permissionLevelName) = 0 Then  [property].SetValue(delegateUser.Permissions, permissionLevel)  Exit For  End If  Next [property]  Dim emailBox As New Mailbox(primaryAddress)  Dim responses As Collection(Of DelegateUserResponse) =  service.AddDelegates(emailBox,  MeetingRequestsDeliveryScope.DelegatesAndSendInformationToMe,  delegateUser)  For Each response As DelegateUserResponse In responses  Console.WriteLine("Add {0} as the delegate to the account {1}:{2}",  delegateAddress, primaryAddress, response.Result)  Console.WriteLine()  Next response  End Sub  --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)

[Working with impersonation by using the EWS Managed API](http://msdn.microsoft.com/en-us/library/dd633680(v=exchg.80).aspx)