# How to Import vCard Files in Office 365 Exchange Online

## Introduction

The vCard file format is supported by many email clients and email services. Now Outlook Web App supports import of single .CSV files only. In this application, we will demonstrate how to import multiple vCard files in Office 365 Exchange Online.

1. Get a single file or all the vCard files in the folder;

2. Read the contact information from the vCard file;

3. Create a new contact and set the properties;

4. Save the contact;

5. Process 2-4 steps for all the vCard files.

## 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 to input a file path or a folder path.



After that, we will create the contacts for the vCard files.



## Using the Code

1. Read information from vCard files.

Before creating the new contact, we need to read and store the contact information from the vCard file. For different entries, we store them in different names.

|  |
| --- |
| -Code block start-  --C# code snippet start--  if (keyName.StartsWith("Names"))  {  String[] names = keyValue.Split(';');  if (names.Length >= 2)  {  contactInfo.Add(ContactSchemaProperties.Surname, names[0]);  contactInfo.Add(ContactSchemaProperties.GivenName, names[1]);  }  }  else if (keyName.StartsWith("FN"))  {  contactInfo.Add(ContactSchemaProperties.DisplayName, keyValue);  }  else if (keyName.StartsWith("ORG"))  {  String[] comDep = keyValue.Split(';');  contactInfo.Add(ContactSchemaProperties.CompanyName, comDep[0]);  contactInfo.Add(ContactSchemaProperties.Companies, comDep[0]);  contactInfo.Add(ContactSchemaProperties.Department, comDep[1]);  }  else if (keyName.StartsWith("TITLE"))  {  contactInfo.Add(ContactSchemaProperties.JobTitle, keyValue);  }  else if (keyName.StartsWith("PHOTO"))  {  ImportContactDetails.ImportPhoto(contactInfo, keyName, reader);  }  else if (keyName.StartsWith("TEL"))  {  ImportContactDetails.ImportTelephone(contactInfo, keyName, keyValue);  }  else if (keyName.StartsWith("ADR"))  {  ImportContactDetails.ImportAddress(contactInfo, keyName, keyValue);  }  else if (keyName.StartsWith("EMAIL"))  {  ImportContactDetails.ImportEmail(contactInfo, keyName, keyValue);  }  --C# code snippet end--  --VB code snippet start--  If keyName.StartsWith("Names") Then  Dim names() As String = keyValue.Split(";"c)  If names.Length >= 2 Then  contactInfo.Add(ContactSchemaProperties.Surname, names(0))  contactInfo.Add(ContactSchemaProperties.GivenName, names(1))  End If  ElseIf keyName.StartsWith("FN") Then  contactInfo.Add(ContactSchemaProperties.DisplayName, keyValue)  ElseIf keyName.StartsWith("ORG") Then  Dim comDep() As String = keyValue.Split(";"c)  contactInfo.Add(ContactSchemaProperties.CompanyName, comDep(0))  contactInfo.Add(ContactSchemaProperties.Companies, comDep(0))  contactInfo.Add(ContactSchemaProperties.Department, comDep(1))  ElseIf keyName.StartsWith("TITLE") Then  contactInfo.Add(ContactSchemaProperties.JobTitle, keyValue)  ElseIf keyName.StartsWith("PHOTO") Then  ImportContactDetails.ImportPhoto(contactInfo, keyName, reader)  ElseIf keyName.StartsWith("TEL") Then  ImportContactDetails.ImportTelephone(contactInfo, keyName, keyValue)  ElseIf keyName.StartsWith("ADR") Then  ImportContactDetails.ImportAddress(contactInfo, keyName, keyValue)  ElseIf keyName.StartsWith("EMAIL") Then  ImportContactDetails.ImportEmail(contactInfo, keyName, keyValue)  End If  --VB code snippet end--  -Code block end- |

For complex information, we’ll use more steps to get the details, such as the Address.

|  |
| --- |
| -Code block start-  --C# code snippet start--  String addrType = keyName.Split(';')[1];  switch (addrType)  {  case "WORK":  {  String[] bussinessAdr = keyValue.Split(';');  if (bussinessAdr.Length >= 1)  {  contactInfo.Add(ContactSchemaProperties.OfficeLocation, bussinessAdr[1]);  }  ContactSchemaProperties[] properties =  {  ContactSchemaProperties.BusinessAddressStreet,  ContactSchemaProperties.BusinessAddressCity,  ContactSchemaProperties.BusinessAddressState,  ContactSchemaProperties.BusinessAddressPostalCode,  ContactSchemaProperties.BusinessAddressCountryOrRegion};  for (Int32 i = 2; i < bussinessAdr.Length; i++)  {  contactInfo.Add(properties[i - 2], bussinessAdr[i]);  }  }  break;  case "HOME":  {  String[] homeAdr = keyValue.Split(';');  ContactSchemaProperties[] properties = {  ContactSchemaProperties.HomeAddressStreet,  ContactSchemaProperties.HomeAddressCity,  ContactSchemaProperties.HomeAddressState,  ContactSchemaProperties.HomeAddressPostalCode,  ContactSchemaProperties.HomeAddressCountryOrRegion};  for (Int32 i = 2; i < homeAdr.Length; i++)  {  contactInfo.Add(properties[i - 2], homeAdr[i]);  }  }  break;  case "POSTAL":  {  String[] postalAdr = keyValue.Split(';');  ContactSchemaProperties[] properties = {  ContactSchemaProperties.OtherAddressStreet,  ContactSchemaProperties.OtherAddressCity,  ContactSchemaProperties.OtherAddressState,  ContactSchemaProperties.OtherAddressPostalCode,  ContactSchemaProperties.OtherAddressCountryOrRegion};  for (Int32 i = 2; i < postalAdr.Length; i++)  {  contactInfo.Add(properties[i - 2], postalAdr[i]);  }  }  break;  default: break;  }  --C# code snippet end--  --VB code snippet start--  Select Case addrType  Case "WORK"  Dim bussinessAdr() As String = keyValue.Split(";"c)  If bussinessAdr.Length >= 1 Then  contactInfo.Add(ContactSchemaProperties.OfficeLocation, bussinessAdr(1))  End If  Dim properties() As ContactSchemaProperties = {  ContactSchemaProperties.BusinessAddressStreet,  ContactSchemaProperties.BusinessAddressCity,  ContactSchemaProperties.BusinessAddressState,  ContactSchemaProperties.BusinessAddressPostalCode,  ContactSchemaProperties.BusinessAddressCountryOrRegion}  For i As Int32 = 2 To bussinessAdr.Length - 1  contactInfo.Add(properties(i - 2), bussinessAdr(i))  Next i  Case "HOME"  Dim homeAdr() As String = keyValue.Split(";"c)  Dim properties() As ContactSchemaProperties = {  ContactSchemaProperties.HomeAddressStreet,  ContactSchemaProperties.HomeAddressCity,  ContactSchemaProperties.HomeAddressState,  ContactSchemaProperties.HomeAddressPostalCode,  ContactSchemaProperties.HomeAddressCountryOrRegion}  For i As Int32 = 2 To homeAdr.Length - 1  contactInfo.Add(properties(i - 2), homeAdr(i))  Next i  Case "POSTAL"  Dim postalAdr() As String = keyValue.Split(";"c)  Dim properties() As ContactSchemaProperties = {  ContactSchemaProperties.OtherAddressStreet,  ContactSchemaProperties.OtherAddressCity,  ContactSchemaProperties.OtherAddressState,  ContactSchemaProperties.OtherAddressPostalCode,  ContactSchemaProperties.OtherAddressCountryOrRegion}  For i As Int32 = 2 To postalAdr.Length - 1  contactInfo.Add(properties(i - 2), postalAdr(i))  Next i  Case Else  End Select  --VB code snippet end--  -Code block end- |

2. Create a new contact.

After getting the contact information, we’ll create a new contact and save it.

|  |
| --- |
| -Code block start-  --C# code snippet start--  foreach (ContactSchemaProperties key in contactInfo.Keys)  {  switch (key)  {  case ContactSchemaProperties.Surname:  newContact.Surname = contactInfo[key];  break;  case ContactSchemaProperties.GivenName:  newContact.GivenName = contactInfo[key];  break;  case ContactSchemaProperties.DisplayName:  newContact.DisplayName = contactInfo[key];  break;  case ContactSchemaProperties.JobTitle:  newContact.JobTitle = contactInfo[key];  break;  case ContactSchemaProperties.Birthday:  {  DateTime birthday;  newContact.Birthday = DateTime.TryParse(contactInfo[key], out birthday) ? (DateTime?)birthday : null;  }  break;  case ContactSchemaProperties.CompanyName:  newContact.CompanyName = contactInfo[key];  break;  case ContactSchemaProperties.Companies:  {  StringList stringList = new StringList();  stringList.Add(contactInfo[key]);  newContact.Companies = stringList;  }  break;  case ContactSchemaProperties.Department:  newContact.Department = contactInfo[key];  break;  case ContactSchemaProperties.EmailAddress1:  newContact.EmailAddresses[EmailAddressKey.EmailAddress1] = contactInfo[key];  break;  case ContactSchemaProperties.EmailAddress2:  newContact.EmailAddresses[EmailAddressKey.EmailAddress2] = contactInfo[key];  break;  case ContactSchemaProperties.EmailAddress3:  newContact.EmailAddresses[EmailAddressKey.EmailAddress3] = contactInfo[key];  break;  case ContactSchemaProperties.BusinessAddressStreet:  case ContactSchemaProperties.BusinessAddressCity:  case ContactSchemaProperties.BusinessAddressState:  case ContactSchemaProperties.BusinessAddressPostalCode:  case ContactSchemaProperties.BusinessAddressCountryOrRegion:  {  if (businessAddressEntry == null)  {  businessAddressEntry = new PhysicalAddressEntry();  }  SetContactDetails.SetAddress(key, contactInfo[key], businessAddressEntry);  }  break;  case ContactSchemaProperties.HomeAddressStreet:  case ContactSchemaProperties.HomeAddressCity:  case ContactSchemaProperties.HomeAddressState:  case ContactSchemaProperties.HomeAddressPostalCode:  case ContactSchemaProperties.HomeAddressCountryOrRegion:  {  if (homeAddressEntry == null)  {  homeAddressEntry = new PhysicalAddressEntry();  }  SetContactDetails.SetAddress(key, contactInfo[key], homeAddressEntry);  }  break;  case ContactSchemaProperties.OtherAddressStreet:  case ContactSchemaProperties.OtherAddressCity:  case ContactSchemaProperties.OtherAddressState:  case ContactSchemaProperties.OtherAddressPostalCode:  case ContactSchemaProperties.OtherAddressCountryOrRegion:  {  if (otherAddressEntry == null)  {  otherAddressEntry = new PhysicalAddressEntry();  }  SetContactDetails.SetAddress(key, contactInfo[key], otherAddressEntry);  }  break;  case ContactSchemaProperties.BusinessPhone:  newContact.PhoneNumbers[PhoneNumberKey.BusinessPhone] = contactInfo[key];  break;  case ContactSchemaProperties.BusinessPhone2:  newContact.PhoneNumbers[PhoneNumberKey.BusinessPhone2] = contactInfo[key];  break;  case ContactSchemaProperties.HomePhone:  newContact.PhoneNumbers[PhoneNumberKey.HomePhone] = contactInfo[key];  break;  case ContactSchemaProperties.HomePhone2:  newContact.PhoneNumbers[PhoneNumberKey.HomePhone2] = contactInfo[key];  break;  case ContactSchemaProperties.MobilePhone:  newContact.PhoneNumbers[PhoneNumberKey.MobilePhone] = contactInfo[key];  break;  case ContactSchemaProperties.Photo:  {  Byte[] picture = Convert.FromBase64String(contactInfo[key]);  newContact.SetContactPicture(picture);  }  break;  default:  break;  }  }  --C# code snippet end--  --VB code snippet start--  For Each key As ContactSchemaProperties In contactInfo.Keys  Select Case key  Case ContactSchemaProperties.Surname  newContact.Surname = contactInfo(key)  Case ContactSchemaProperties.GivenName  newContact.GivenName = contactInfo(key)  Case ContactSchemaProperties.DisplayName  newContact.DisplayName = contactInfo(key)  Case ContactSchemaProperties.JobTitle  newContact.JobTitle = contactInfo(key)  Case ContactSchemaProperties.Birthday  Dim birthday As Date  newContact.Birthday =  IIf(Date.TryParse(contactInfo(key), birthday), birthday, Nothing)  Case ContactSchemaProperties.CompanyName  newContact.CompanyName = contactInfo(key)  Case ContactSchemaProperties.Companies  Dim stringList As New StringList()  stringList.Add(contactInfo(key))  newContact.Companies = stringList  Case ContactSchemaProperties.Department  newContact.Department = contactInfo(key)  Case ContactSchemaProperties.EmailAddress1  newContact.EmailAddresses(EmailAddressKey.EmailAddress1) = contactInfo(key)  Case ContactSchemaProperties.EmailAddress2  newContact.EmailAddresses(EmailAddressKey.EmailAddress2) = contactInfo(key)  Case ContactSchemaProperties.EmailAddress3  newContact.EmailAddresses(EmailAddressKey.EmailAddress3) = contactInfo(key)  Case ContactSchemaProperties.BusinessAddressStreet,  ContactSchemaProperties.BusinessAddressCity,  ContactSchemaProperties.BusinessAddressState,  ContactSchemaProperties.BusinessAddressPostalCode,  ContactSchemaProperties.BusinessAddressCountryOrRegion  If businessAddressEntry Is Nothing Then  businessAddressEntry = New PhysicalAddressEntry()  End If  SetContactDetails.SetAddress(key, contactInfo(key), businessAddressEntry)  Case ContactSchemaProperties.HomeAddressStreet,  ContactSchemaProperties.HomeAddressCity,  ContactSchemaProperties.HomeAddressState,  ContactSchemaProperties.HomeAddressPostalCode,  ContactSchemaProperties.HomeAddressCountryOrRegion  If homeAddressEntry Is Nothing Then  homeAddressEntry = New PhysicalAddressEntry()  End If  SetContactDetails.SetAddress(key, contactInfo(key), homeAddressEntry)  Case ContactSchemaProperties.OtherAddressStreet,  ContactSchemaProperties.OtherAddressCity,  ContactSchemaProperties.OtherAddressState,  ContactSchemaProperties.OtherAddressPostalCode,  ContactSchemaProperties.OtherAddressCountryOrRegion  If otherAddressEntry Is Nothing Then  otherAddressEntry = New PhysicalAddressEntry()  End If  SetContactDetails.SetAddress(key, contactInfo(key), otherAddressEntry)  Case ContactSchemaProperties.BusinessPhone  newContact.PhoneNumbers(PhoneNumberKey.BusinessPhone) = contactInfo(key)  Case ContactSchemaProperties.BusinessPhone2  newContact.PhoneNumbers(PhoneNumberKey.BusinessPhone2) = contactInfo(key)  Case ContactSchemaProperties.HomePhone  newContact.PhoneNumbers(PhoneNumberKey.HomePhone) = contactInfo(key)  Case ContactSchemaProperties.HomePhone2  newContact.PhoneNumbers(PhoneNumberKey.HomePhone2) = contactInfo(key)  Case ContactSchemaProperties.MobilePhone  newContact.PhoneNumbers(PhoneNumberKey.MobilePhone) = contactInfo(key)  Case ContactSchemaProperties.Photo  Dim picture() As Byte = Convert.FromBase64String(contactInfo(key))  newContact.SetContactPicture(picture)  Case Else  End Select  Next key  --VB code snippet end--  -Code block end- |

At last we’ll add the address entry into the contact and save it.

|  |
| --- |
| -Code block start-  --C# code snippet start--  if (businessAddressEntry != null)  {  newContact.PhysicalAddresses[PhysicalAddressKey.Business] = businessAddressEntry;  }  if (homeAddressEntry != null)  {  newContact.PhysicalAddresses[PhysicalAddressKey.Home] = homeAddressEntry;  }  if (otherAddressEntry != null)  {  newContact.PhysicalAddresses[PhysicalAddressKey.Other] = otherAddressEntry;  }  newContact.FileAsMapping = FileAsMapping.GivenNameSpaceSurname;  newContact.Save(WellKnownFolderName.Contacts);  --C# code snippet end--  --VB code snippet start--  If businessAddressEntry IsNot Nothing Then  newContact.PhysicalAddresses(PhysicalAddressKey.Business) = businessAddressEntry  End If  If homeAddressEntry IsNot Nothing Then  newContact.PhysicalAddresses(PhysicalAddressKey.Home) = homeAddressEntry  End If  If otherAddressEntry IsNot Nothing Then  newContact.PhysicalAddresses(PhysicalAddressKey.Other) = otherAddressEntry  End If  newContact.FileAsMapping = FileAsMapping.GivenNameSpaceSurname  newContact.Save(WellKnownFolderName.Contacts)  --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)

[Contact class](http://msdn.microsoft.com/en-us/library/exchange/microsoft.exchange.webservices.data.contact(v=exchg.80).aspx)