

Content Manager

Software Version 23.4

Zero Foot Print integration

opentext™

Document Release Date: July 2025
Software Release Date: July 2025

Legal notices

Copyright 2008-2025 Open Text

The only warranties for products and services of Open Text and its affiliates and licensors ("Open Text") are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Except as specifically indicated otherwise, this document contains confidential information and a valid license is required for possession, use or copying. If this work is provided to the U.S. Government, consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Documentation updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

Support

Visit the [MySupport portal](#) to access contact information and details about the products, services, and support that OpenText offers.

This portal also provides customer self-solve capabilities. It gives you a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the MySupport portal to:

- View information about all services that Support offers
- Submit and track service requests
- Contact customer support
- Search for knowledge documents of interest
- View software vulnerability alerts
- Enter into discussions with other software customers
- Download software patches
- Manage software licenses, downloads, and support contracts

Many areas of the portal require you to sign in. If you need an account, you can create one when prompted to sign in.

Contents

- Introduction 4
 - Components 4
 - Requirements 4
- Service API 5
 - Service API configuration 5
 - Office integration configuration 5
 - Service API authentication permissions 5
 - Service API authentication application ID 6
 - Service API authorized client applications 7
- Manifests 8
 - Browser Cookies 8
- EmailLink 9
- Content Manager menu 10
- Nested App Authentication for Office Add-ins 11
 - New setup script to create new Azure applications 11

Introduction

The Content Manager Zero FootPrint (ZFP) Office integration document enables you to integrate Content Manager with Zero FootPrint (ZFP) Office applications, Teams, and Outlook.

Components

The Zero Foot Print (ZFP) Office integration is made up of a number of components. You can configure the components depending on the level of functionality required. The components are:

- Service API
- Office 365 application manifests
- EmailLink

Requirements

The components listed above work together to provide the complete Zero Foot Print integration. If you need only Outlook integration, then configure the Service API, EmailLink and Outlook manifest. The other components, Office manifest, is required for Word, Excel, and PowerPoint integration.

Service API

The Service API must be installed and must have a valid HTTPS certificate, any HTTP errors will cause the Office integration to fail.

For more details on installing and configuring the Service API, see *Content Manager Service API* document.

The Service API performs three functions within the integration:

- hosts the web application that is embedded in Office as the add-in (at /CMServiceAPI/office/index.html),
- provides the backend services to which the add-on communicates when reading from or writing to Content Manager, and
- provides services that act as a proxy between the add-in and OneDrive for reading from and writing to OneDrive.

Service API configuration

Office integration configuration

The Service API must be provided with the GUID and the version found in the ID element of the Office add-in [manifest](zero_manifests). This GUID is used when a document is first opened from Content Manager to notify the add-in that the Content Manager add-in should be auto-opened when the document is opened.

To add this GUID,

1. Edit the **hptrim.config** file.
2. Add the **officeIntegrationguid** element below the setup element

For example,

```
<setup databaseId="21" />  
<officeIntegrationguid="5d7bd8ba-11b0-99b9-98ab-99fadd99a99d" version="1.0.0.0"/>
```

Service API authentication permissions

The Service API must be configured to use OpenID Connect authentication as described in the *Content Manager OpenID Connect authentication setup* document. The following new permissions must be assigned to the Azure AD application to support OneDrive access:

- Delegated:
 - Files.Read.All
 - Files.ReadWrite.All
- Application:
 - Files.Read.All
 - Files.ReadWrite.All
 - Office 365 Exchange Online - full_access_as_app

To assign these permissions,

1. Open the Azure AD application used for authenticating the Web Client.
2. Go to **API permissions** and select **Add a permission**.
3. Select **Microsoft Graph**.
4. Choose either **Delegated** or **Application** permissions.
5. Find the permission and add it.
6. Once you have added all the permissions, select **Grant admin consent** to consent to all permissions.

NOTE: The Office 365 Exchange Online permission is not part of the OneDrive permissions, to find it, perform the following steps:

- a. Click **Add a permission** and then select **API's my organisation uses**.
- b. Find the **Office 365 Exchange Online** in the listed.
- c. Add the **Application** permission.

Service API authentication application ID

The Office integration requires that the application ID URI in the Azure AD app include the domain name of your Service API server. This must be adjusted before installing the Office integration manifests.

To adjust the Application ID URI,

1. Open the Azure App referenced from your **ServiceAPI hptrim.config** file.
2. Go the **Expose an API** page.
3. Edit the **Application ID URI** field.

By default the Application ID URI will look something like `api://2d89cb2b-3cb7-4c21-807e-029fdbfe74a7`,

4. Edit the Application ID URI to include the name of your Service API server. For example `api://myserver.com/2d89cb2b-3cb7-4c21-807e-029fdbfe74a7`

5. Once you have modified the Application ID URI, you will need to update the **appldURI** property (case sensitive) in the **ServiceAPI hprim.config** file.

Service API authorized client applications

The ZFP Office integration relies on Single Sign On from the Microsoft add-in framework. To allow this to work, we need to register our add-in. This is done in the **Expose an API** section of the Azure App created previously.

To register your add-in,

1. Add a Scope.
2. Name it `access_as_user`
3. Fill in the fields with appropriate values. For example,
 - **Admin consent display name:** Office can act as the user.
 - **Admin consent description:** Enable Office to call the add-in's web APIs with the same rights as the current user.
 - **User consent display name:** Office can act as you.
 - **User consent description:** Enable Office to call the add-in's web APIs with the same rights that you have.
4. Save the scope.
5. Add six client applications, each time selecting the scope you just added, the Client ID for each of these are:
 - d3590ed6-52b3-4102-aeff-aad2292ab01c
 - 57fb890c-0dab-4253-a5e0-7188c88b2bb4
 - bc59ab01-8403-45c6-8796-ac3ef710b3e3
 - ea5a67f6-b6f3-4338-b240-c655ddc3cc8e
 - 93d53678-613d-4013-afc1-62e9e444a0a5
 - 08e18876-6177-487e-b8b5-cf950c1e598c

Manifests

The manifest's purpose is to notify Office 365 of the presence of your add-in. Manifest files must be customised before upload to your Office 365 Admin.

To prepare your manifests,

1. Download the sample manifests ([office](#), [outlook](#)).
2. [Generate](#) a new Guid for each manifest (if you are installing both Outlook and Office integrations).

NOTE: If this is an upgrade, use the Guid of the originally installed manifest. Ensure the **Version** element contains a higher number than the original.

3. Replace [MANIFESTGUID] in each manifest with a valid Guid.
4. Set the GUID from the Office manifest in the **officeIntegration** element in the **hptrim.config** file (see above).
5. Assuming the Service API is in the pattern <https://yourserver.com/CMServiceAPI>, do a search and replace of each manifest, replacing [SERVICEAPIURL] with your URL.
6. Replace [DOMAIN] with your web server domain name.
7. In the **WebApplicationInfo** element of the manifest, replace [APPCLIENTID] with the Application ID from Azure and [APPIDURI] with the Application ID URI from Azure.

The name of the add-in that will be displayed in the task pane title (and in the context menu for Outlook) is embedded in the manifest. By default, it is **Content Manager** but this can be changed, if desired.

8. Go to the [Add-ins page](#) in Office 365 Admin and deploy each manifest.

NOTE: The setup script used for the Microsoft Teams integration also generates manifests for Office, this may be simpler than following the manual steps above. For details, see *Content Manager Microsoft Teams Integration* document.

Browser Cookies

It may be that the Office add-in does not display after it has been uploaded. To resolve this, ensure that all cookie blocking is disabled in the web browser.

EmailLink

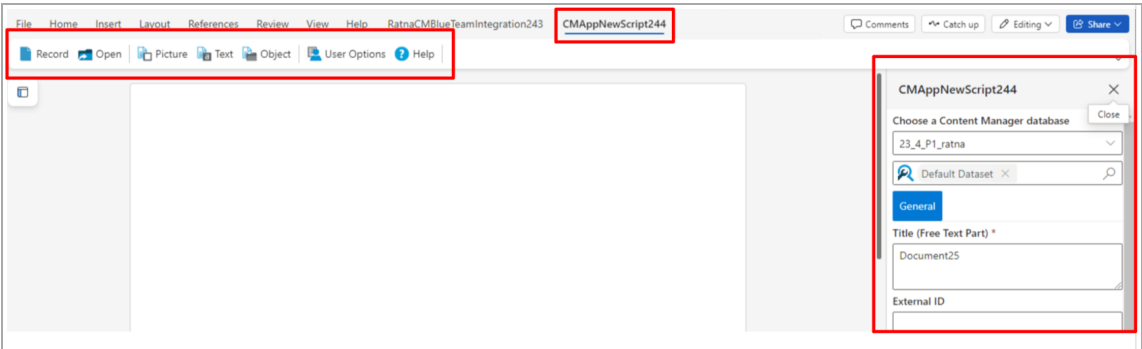
EmailLink serves two purposes in the Outlook/Office integration:

1. File email from Linked Folders in the Outlook integration, and
2. Check documents in from Office applications when **Check In on close** is selected.

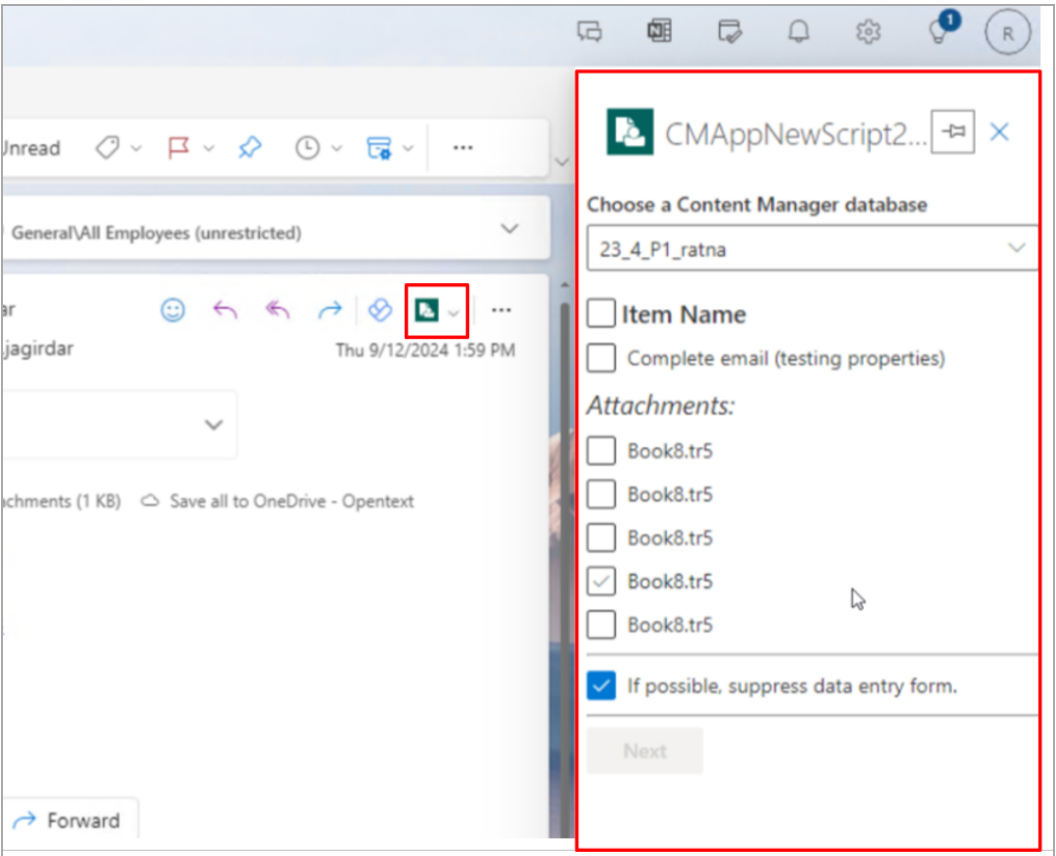
EmailLink should be installed and there must be a Microsoft authentication configuration for your Office 365 domain that is of type OAuth and has both Email and Drive selected (if both Office and Outlook integration is required).

Content Manager menu

After the successful configuration, when you open Office or Outlook application, the Content Manager menu will be available on the menu bar.



Content Manager menu in Office



Content Manager menu in Outlook

Nested App Authentication for Office Add-ins

Due to recent changes on the legacy Exchange tokens being turned off by Microsoft and providing new Nested App Authentication for Office Add-ins, you may not have been able to operate the ZFP Outlook integration (existing or new).

New setup script to create new Azure applications

From 24.3 onwards, as part of Content Manager installation, a new setup script, **az_install.ps1**, is available at the installation location (for example, **C:\Program Files\Micro Focus\Content Manager\Service_API\office\AzureScript**) to create new Azure app for Content Manager MS Teams and ZFP Office/Outlook integrations.

The administrator must use/run the setup script, **az_install.ps1**, to create new Azure app.

For existing Content Manager Azure App users,

For the existing ZFP Outlook integrations, perform the following additional configurations:

1. Adding the Single-page application

- a. Go to www.portal.azure.com and click **App registrations**.
- b. In the **Owned applications** tab, click on your registered app.
- c. In the **Overview** section, click **All Microsoft account users** for the Supported account types. The Platform configurations page is displayed.
- d. Click **Add a platform** and then click **Single-page application** from the right panel. The **Configure single-page application** panel is displayed.
- e. Enter the Redirect URI of the application in the text box and click **Configure**.

IMPORTANT: Make sure to add the following two redirect URIs adhering to the naming convention:

- i. `https://yourServiceAPIname/office/index.html`.

For example, `https://localhost/myserviceapi/office/index.html`.

- ii. `brk-multihub://your-add-in-domain`

For example, `brk-multihub://localhost:3000`, if you are testing your add-in on port 3000 on the localhost server.

2. Adding the permissions:

- a. In your registered application, navigate to the **API permissions** section and click **Add a permission**.
- b. In the **Request API permissions** panel, click **Microsoft Graph** and then click **Delegated permissions**.
- c. Search and select the following permissions and click **Add permissions**.

- **User.Read**

3. Updating the config.js file

- a. Navigate to the **Content Manager installation folder > Service_API > office**. For example, C:\Program Files\Micro Focus\Content Manager\Service_API\office.
- b. Open the **config.js** in a text editor and add the **APPLICATION_ID** and **AUTHORITY** parameters. For example,

```
...  
DEFAULT_AUTO_OPEN: true,  
SEARCH_CLAUSES: {  
  Record: [],  
  SavedSearch: ["top"],  
},  
APPLICATION_ID: "XfXffXac-XbaX-XXXc-XeXd-XXdXXeXXXXXX",  
AUTHORITY: "https://login.microsoftonline.com/ZfXffXab-YbcX-XXXc-YeZd-  
XZdXXeXXZZYYX"  
};
```

IMPORTANT: Make sure to use `https://login.microsoftonline.com/common` as **AUTHORITY** if you have multiple tenants. Otherwise, use `https://login.microsoftonline.com/(tenant ID)` as **AUTHORITY**.

- c. Save and close the file.

IMPORTANT: For Shared mailbox and delegations

If you have existing Content Manager Azure App with Shared mailbox, you have to manually add the following permissions:

- **Mail.Read.Shared**
- **Mail.ReadWrite.Shared**

For detailed steps, see *Configuring additional permissions* section in *Content Manager Outlook Online Integration* help.