# Office AMS: Bulk Upload user Profile pictures

|  |  |
| --- | --- |
| Summary: | Applies to: |
| This sample application creates a YouTube | Outlook Web Access  Outlook Client |
| Solution: | Core.MailApps, version 1.0 |
| Author: | Suman Chakrabarti (Microsoft) |
| //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // THIS CODE IS PROVIDED \*AS IS\* WITHOUT WARRANTY OF  // ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING ANY  // IMPLIED WARRANTIES OF FITNESS FOR A PARTICULAR  // PURPOSE, MERCHANTABILITY, OR NON-INFRINGEMENT.  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | |

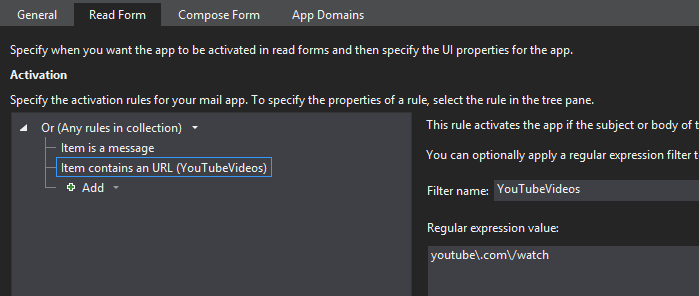
# Scenario: YouTube app for Exchange 2013/Online Mailbox

Outlook is the one application every enterprise user opens every day. Exchange 2013 provides a javascript-based interface for building apps that integrate into the Exchange experience and via the Outlook Web Access and within the Outlook client.

The documentation below describes how we can easily create an application that will pull information out of an email to extend the functionality of Outlook and Outook Web Access:

## Office Mail App Manifest

The Office App Manifest is used to describe to the Exchange server what the app is supposed to look for. In the sample app, the app looks for the following:



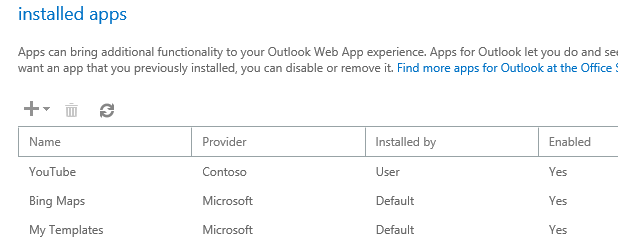
The regular expression for this URL search will either yield results or will not yield them. If it does yield a result, the app button will appear in the message. The data for this app will be available by using the following methods:

* mailitem.getEntities()
* mailitem.getEntitiesByType()
* mailitem.getFilteredEntitiesByName()
* mailitem.getRegExMatches()
* mailitem.getRegExMatchesByName()

These methods will be the point for making the content easily available within the application.

### Configuration

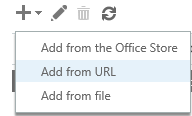
Office mail apps require that the tenant or the individual subscribe to a manifest file, which can be hosted at a web site location or added directly to the user’s apps. **Settings->Options->Apps** takes you to the **Installed Apps** page which looks like this when the app is deployed:

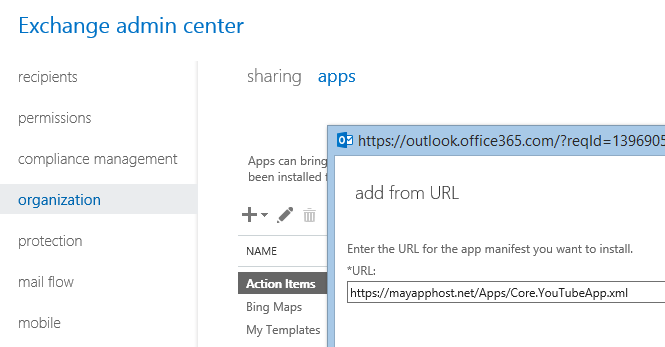


The manifest simply points back to the host URL where a web site provides the script for running the application which runs in an iframe. You can have many apps under the Visual Studio Project and these will be deployed to your specific mailbox In the solution during build. The manifest file for the YouTube application is located under **Core.MailApps** and looks like this:



When building this project, there is a post-build event which runs the **UpdateAppUrl.ps1** script. The file will update the *SourceLocation* in all manifest files *~remoteAppUrl* property in the project with whatever is set in the $hostUrl property in the script. This is the URL that will be used at the user or tenant level for adding an app. Below is a screenshot of the tenant-level administration page where the app is being added.





## Web Application

The web application is not unlike any other web application. It can be a .NET application, Node.js, PHP, Java, etc. The point is that it’s limitless what you can create here, the web page is within the iframe. Note that the Office.css is included in the project to ensure that the UI follows the inteface of Outlook and Outlook Web Access.

The web project contains the App.css and App.js for providing application-level scripts which may be used across many applications within the web. Note that this app could simply be hosted in the same web site as the entire application. The **YouTube** folder contains a specific page which is rendered in the iframe when the app is initialized.

### Project Structure

When you create a Mail app, you are given the choice to create an app that reads the contents of your email or that writes/composes an email. In the YouTube sample, the app simply reads the email, finds URLs which match the youtube watch format, matches them with a regular expression and makes them available to the app. The final result looks like so:



# References

For more information on developing mail apps, visit the Office Mail Apps development center on MSDN:  
<http://msdn.microsoft.com/en-us/library/office/fp161135(v=office.15).aspx>