DIWUG

Dutch Information Worker User Group

2006 - 2016

10 Years Anniversary Edition

SharePoint®

Introducing OfficeDev Patterns and Practices

Information management Office 365

The Insider Security Threat, and How SharePoint / Office 365 Measures Up

A new world of Front-end development

Office 365 collaboration tools: what to use for what?

Practical guidance for deploying high trust add-ins on-premises

Developing Office 365 apps with the Microsoft Graph and Office UI Fabric

Welcome back old friend

Everything you need to know about cloud hybrid search

Using PowerApps to create line of business solutions

#17
June 2016

Contents

Colofon

DIWUG SharePoint eMagazine
Nr. 17, June 2016

Publisher:
Stichting Dutch Information Worker User Group (DIWUG)
http://www.diwug.nl

Editors:
Marianne van Wanrooij
marianne@diwug.nl
Mirjam van Olst
mirjam@diwug.nl
Maarten Eekels
maarten@diwug.nl

Special thanks to:
All authors and sponsors!

Design and layout:
Barth Sluyters
barthsluyters@telfort.nl

Editors note

DIWUG community 10 years!

This year we celebrate our 10 anniversary! 10 years ago Mart Muller had an idea to share knowledge about his work with other SharePoint developers. He arranged a room at the Microsoft Innovation Center in Baarn and emailed all developers in his network. He gave a session (and I have to admit, I can't remember the subject) and after this session he asked if there were others who would like to help him set up the DIWUG. I volunteered immediately.

Starting the DIWUG wasn't easy. Finding speakers and sponsors was hard and our network was still small. And we didn't have a real plan. We started organizing an event when we had the time. After 6 months Mirjam van Olst joined our team. And we decided that we would organize events more frequently. DIWUG was voluntary but not optional! Every month we tried to organize an evening for the community with different audiences - developers, end users or IT-pros. As we organize events more frequently our community started to grow. Working together with the Dutch Software Development Network gave us insight on how to expand our community even further by not only organizing events but also publishing a magazine. The DIWUG SharePoint Magazine was born!

Next step was organizing a SharePoint Saturday! Together with the Office365 User Group, this year we have the 6th SPS in the Netherlands with approximately 300 delegates and international speakers.

This year Maarten Eekels joined our team, and he has been a great help, since we still do this next to our daily job!

Now every year we organize 9 DIWUG events, 1 SharePoint Saturday and publish 3 to 4 magazines for our 1500 members.

We like to thank all of our loyal members, speakers, writers and our sponsors who made it possible to do this for over 10 years! It has been a real joy and we're looking forward to next decade!

Enjoy this issue of the DIWUG SharePoint Magazine as we enjoyed putting it together!

Marianne van Wanrooij

Editor DIWUG SharePoint Magazine

This issue is sponsored by

When using the Adobe® Acrobat® version all sponsor information is hyper linked.

Introducing OfficeDev Patterns and Practices

by Paolo Pialorsi

The OfficeDev Patterns and Practices (PnP) initiative (http://aka.ms/OfficeDevPnP) was born in 2013 by the efforts of some Microsoft internals people - mainly consultants - focused on supporting big customers and enterprises in transforming their Full Trust Code (FTC) solutions into the new Add-In Model. Pretty soon they realized that sharing samples, common patterns, and guidance articles would have been the best way to spread the knowledge about the new development model introduced by the cloud and by Microsoft Office 365. Thus, the initiative became a community and open source project, to which both Microsoft and common external people can contribute, and that right now is one of the biggest community projects around a Microsoft technology.

The initial focus was SharePoint, but step by step the target moved to the entire Microsoft Office 365 ecosystem. Right now, the project is managed by a Core Team made of people both from Microsoft and from the community, and it is focused on some main areas and topics:

- PnP Core Library: it is a .NET assembly which provides useful types and extensions methods to easily develop with the new SharePoint add-in model, targeting both SharePoint on-premises and SharePoint Online. It is open source and is available on GitHub: http://aka.ms/OfficeDevPnPSitesCore.
- PnP Remote Provisioning Engine: it is a component of the PnP Core Library and allows to do remote provisioning via Client Side Object Model (CSOM) of artifacts, targeting both SharePoint on-premises and SharePoint online. It also allows to save a site as a template, which can be applied onto another site.
- PnP PowerShell Extensions: it is a set of PowerShell cmdlets that allow to remotely control and manage SharePoint on-premises and SharePoint Online. It is still open source, and is available at the following URL: http://aka.ms/OfficeDevPnPPowerShell.
- PnP Samples: a rich set of code samples, which allow people to understand how to do things in the proper way. It is available at the following URL: http://aka.ms/OfficeDevPnPSamples.
- PnP Yammer Group: it is one of the biggest technical groups in the Yammer network for Microsoft Office 365. It is available at the following URL: http://aka.ms/OfficeDevPnPYammer.
- PnP Guidance: it is a collection of guidance articles authored by the community and validated by Microsoft technical people. It is available at the following URL: http://aka.ms/OfficeDevPnPGuidance.
- PnP Partner Pack: it is a sample solution that leverages the PnP Core Library and the PnP Remote Provisioning Engine to provide a starting point for creating an enterprise-level site provisioning solution for Microsoft Office 365. It is available at the following URL: http://aka.ms/OfficeDevPnPPartnerPack.

In this article we will briefly introduce the PnP Core Library and the PnP Remote Provisioning Engine.

PnP Core Library

The new add-in model introduced in Microsoft SharePoint 2013 moved the developers' focus from the FTC development model to the client-side development through CSOM. However, quite often using CSOM to achieve common tasks like creating a list or a library, adding an item to a target list, managing artifacts and contents of a site in general is not always easy and straight forward.

For example, consider the code excerpt in Listing 1, which is a very minimal sample about how to add to a site a list instance based on the Contacts template, by using the standard CSOM syntax:

```
using (var context = new ClientContext(targetUrl)) {
  context.Credentials = new SharePointOnlineCredentials(userName, password);
  Web web = context.Web;
  ListCreationInformation newList = new ListCreationInformation() {
    Title = "Contacts via CSOM",
    Url = String.Format("lists/{0}", Guid.NewGuid()),
    TemplateType = (Int32)ListTemplateType.Contacts,
    QuickLaunchOption = QuickLaunchOptions.On,
  };

  web.Lists.Add(newList);
  context.ExecuteQuery();
}
```

Listing 1: Code excerpt to create a list of contacts using standard CSOM.

The same result, using the PnP Core Library, can be achieved with the code excerpt illustrated in Listing 2::

```
using (var context = new ClientContext(targetUrl)) {
  context.Credentials = new SharePointOnlineCredentials(userName, password);
  context.Web.CreateList(ListTemplateType.Contacts,
    "Contacts via PnP", false,
    urlPath : String.Format("lists/{0}", Guid.NewGuid()));
}
```

Listing 2: Code excerpt to create a list of contacts using the PnP Core Library.

As you can see, by using the CreateList extension method, you can create a list instance with just one short and self-explanatory statement. For the sake of simplicity, this is a very short example, but more in general you should consider that the PnP Core Library provides hundreds of extensions methods to accomplish many of the most frequently needed actions. There are also some interesting webcasts available on Channel 9 (http://aka.ms/OfficeDevPnPVideos) about what can be done with PnP.

Moreover, you should consider that within its internal implementation the PnP Core Library handles - and solves for you - the most common issues that could happen while interacting with SharePoint via CSOM, especially in the cloud context.

For example, the CSOM library provides the ExecuteQuery method for the ClientContext type. However, while consuming SharePoint Online via CSOM or REST, there are some throttling rules that can be applied by the cloud platform and your requests could be blocked. If you use the standard ExecuteQuery method, you should manually handle such situations and provide a retry logic. Luckily, with the PnP Core Library you have an ExecuteQueryRetry extension method that automatically handles the retry logic for you. In Listing 3, you can see a code excerpt that illustrates how to use the ExecuteQueryRetry extension method.

```
using (var context = new ClientContext(targetUrl)) {
  context.Credentials = new SharePointOnlineCredentials(userName, password);
  Web web = context.Web;
  context.Load(web, w => w.Title);
  context.ExecuteQueryRetry();
}
```

Listing 3: Code excerpt using the ExecuteQueryRetry extension method of the PnP Core Library.

As you can see, the ExecuteQueryRetry method extends the ClientContext type and it is optimized for the Microsoft Office 365 throttling rules. However, you can even customize its behavior by providing a couple of optional arguments (retryCount and delay) that should be self-explanatory.

The list of extension methods and types provided by the PnP Core Library could be really long, but it is way more better to start downloading the library through NuGet (search for SharePointPnP, like illustrated in Figure 1) and start playing with it.

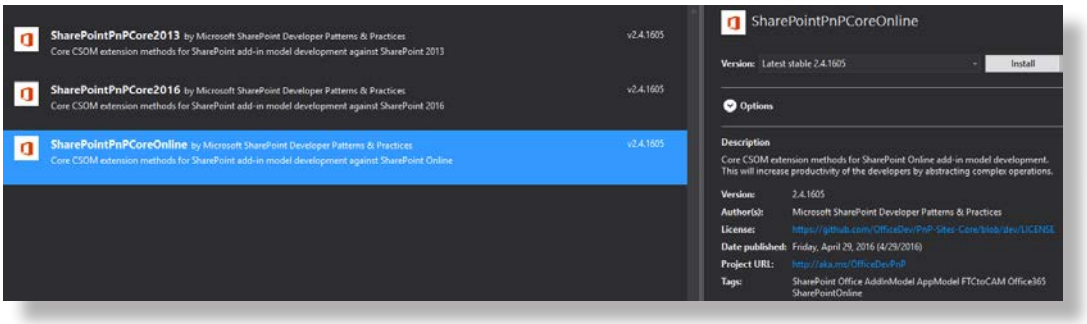


Figure 1: The SharePointPnP Core Library package in NuGet, within Microsoft Visual Studio 2015.

As you can see there are three flavors available for the library. The SharePointPnPCore2013 package targets SharePoint 2013 on-premises and uses the CSOM library for SharePoint 2013 on-premises. The SharePointPnPCore2016 package targets SharePoint 2016 on-premises and uses the CSOM library for SharePoint 2016. Lastly, the SharePointPnPCoreOnline package targets SharePoint Online and leverages the latest CSOM library for SharePoint Online. Depending on your target platform you will have to download the proper package. Nevertheless, all the packages provide almost the same set of capabilities, except for those functionalities that are available only on the cloud.

PnP Remote Provisioning Engine

Another amazing feature included in the PnP Core Library is the so called “PnP Remote Provisioning Engine”. Doing Remote Provisioning means using CSOM to provision artifacts, instead of using the “old school” CAML/XML-based feature framework. In fact, while transforming FTC solutions, WSP packages, and Sandboxed Solutions into the new add-in model, you should also approach provisioning of artifacts in a more maintainable manner. In fact, using pure CSOM allows you to control by code the provisioning and the versioning of artifacts. Moreover, this is the officially recommended option by Microsoft engineering, since CAML/XML-based provisioning will cause maintenance challenges with the evolving templates or definitions. Nevertheless, doing all the provisioning manually and by writing CSOM-based code could be a long and painful task.

Luckily, the PnP Core Team and the whole OfficeDev PnP community have built an engine, which is part of the PnP Core Library and which leverages the PnP Core Library extensions, that allows to easily provision artifacts. Moreover, the PnP Remote Provisioning Engine allows you even to model your artifacts within the web browser, by using a prototype or a model site, and to extract the designed artifacts into a template file, which can then be applied to any target site.

The overall goal of the PnP Remote Provisioning Engine is to make it simple what is really useful and common when provisioning sites and artifacts. The provisioning template can be created in memory, using a domain model that is defined within the PnP Core Library, or as already stated can be persisted as a file. In the latter scenario, out of the box the file can be an XML file, based on a community defined XML Schema (https://github.com/OfficeDev/PnP-Provisioning-Schema), or it can be a JSON file. By default, a template can be read from or written to a file system folder, a document library in SharePoint, or a container in the Azure Blob Storage. However, from an architectural perspective you can implement your own template formatter and your custom persistence provider, in order to save or load a template with whatever format and persistence storage you like.

From a developer perspective, extracting a template from an already existing site, which can be a model site that you designed in the browser for that specific purpose, can be accomplished like illustrated in Listing 4.

```
using (var context = new ClientContext(sourceUrl)) {
using (var context = new ClientContext(sourceUrl)) {
  context.Credentials = new SharePointOnlineCredentials(userName, password);
  Web web = context.Web;

  // Configure the XML file system provider
  XMLTemplateProvider provider =
    new XMLFileSystemTemplateProvider(
      String.Format(@"{0}\..\..\",
      AppDomain.CurrentDomain.BaseDirectory), "");

  // Configure the template extraction behavior (optional step)
  ProvisioningTemplateCreationInformation ptci =
    new ProvisioningTemplateCreationInformation(web);
  ptci.MessagesDelegate += delegate (string message,
    ProvisioningMessageType messageType) {
      Console.WriteLine("{0} - {1}", messageType, messageType);
  };
  ptci.ProgressDelegate += delegate (string message, int step, int total) {
    Console.WriteLine("{0:00}/{1:00} - {2}", step, total, message);
  };
  ptci.PersistComposedLookFiles = true;
  ptci.IncludeAllTermGroups = true;
  ptci.IncludeSiteGroups = true;
  ptci.FileConnector = provider.Connector;

  // Extract the current template
  ProvisioningTemplate template = web.GetProvisioningTemplate(ptci);

  // And save it on the file system
  provider.SaveAs(template, "Provisioning-Template.xml");
}
```

Listing 4: Code excerpt to extract a provisioning template from an existing site.

The key point of the code excerpt illustrated in Listing 4 is the invocation of the GetProvisioningTemplate extension method, which extends the Web type of CSOM and gives back an object graph that is the in memory provisioning template. By calling the SaveAs method of any of the available providers, you will be able to store the template through a persistence provider. In Listing 4, the provisioning template is stored on the file system, within an XML file.

After having created a provisioning template, you can load it and apply it onto a target site simply by using the syntax illustrated in Listing 5.

```
using (var context = new ClientContext(targetUrl)) {
  context.Credentials = new SharePointOnlineCredentials(userName, password);
  Web web = context.Web;

  // Configure the XML file system provider
  XMLTemplateProvider provider =
    new XMLFileSystemTemplateProvider(
      String.Format(@"{0}\..\..\",
       AppDomain.CurrentDomain.BaseDirectory), "");

  // Load the template from the XML stored copy
  ProvisioningTemplate template =
    provider.GetTemplate("Provisioning-Template.xml");
  template.Connector = provider.Connector;

  // Configure how the template provisioning will behave (optional step)
  ProvisioningTemplateApplyingInformation ptai =
    new ProvisioningTemplateApplyingInformation();
  ptai.MessagesDelegate += delegate (string message,
    ProvisioningMessageType messageType) {
      Console.WriteLine("{0} - {1}", messageType, messageType);
  };
  ptai.ProgressDelegate += delegate (string message, int step, int total) {
    Console.WriteLine("{0:00}/{1:00} - {2}", step, total, message);
  };

  // Apply the template to the target site
  web.ApplyProvisioningTemplate(template, ptai);
}
```

Listing 5: Code excerpt to extract a provisioning template from an existing site.

As you can see, the template application happens under the cover of the ApplyProvisioningTemplate extension method, which again extends the Web type of CSOM.

It is fundamental and really interesting to notice that the template application is based on the initial site template, which has been used to create the site before customizing its artifacts, as well as on the latest applied templates and customizations. In fact, when you extract a template from a site, under the cover the PnP Provisioning Engine will extract only the differences between the original site template (for example STS#0 for a "Team Site") and the current site outline.

Moreover, when you apply a template to a site, the engine will do the delta handling, basically upgrading the target site artifacts to the applied template. This process also updates sites already provisioned with the PnP Provisioning Engine. Thus, the PnP Provisioning Engine supports the full life cycle management of sites from initial configuration to updates. By design, and for security reasons the engine does not delete any artifact, while applying a template, but any addition or configuration changes to the already existing artifacts will be applied to the target.

This way, you can use the PnP Remote Provisioning Engine as a tool for provisioning sites and keeping their artifacts in sync with a reference model. This is a very useful capability, especially when you have multiple environments like development, testing/pre-production, production, or when you want to distribute your solutions to multiple customers.

From a capabilities perspective, the PnP Remote Provisioning Engine can handle many of the most useful kind of artifacts. At the time of writing (December 2015), here are the main features and capabilities:

- Regional Settings
- Multiple UI Languages
- Audit Settings
- Site Security (users, groups)
- Taxonomies
- Site Columns
- Content Types
- List instances
- Lookup Fields
- Custom Actions
- Features
- Composed Looks
- Property bag entries
- Workflows (2013 model) defined in SharePoint Designer 2013
- Files, Pages, WikiPages, and WebParts
- Publishing contents (Page Layouts and Master Pages)
- Search Settings

Nevertheless, keep in mind that the project is open source and it is based on community efforts. Thus, it is growing on a daily basis following the real needs of the Office 365 developers' community.

Lastly, consider that you can benefit from all the capabilities provided by the PnP Remote Provisioning Engine also without writing any line of code and simply by using some PowerShell scripting. It will suffice to download the PnP PowerShell extensions (http://aka.ms/OfficeDevPnPPowerShell) and using the Get-SPOProvisioningTemplate and the Apply-SPOProvisioningTemplate cmdlets.

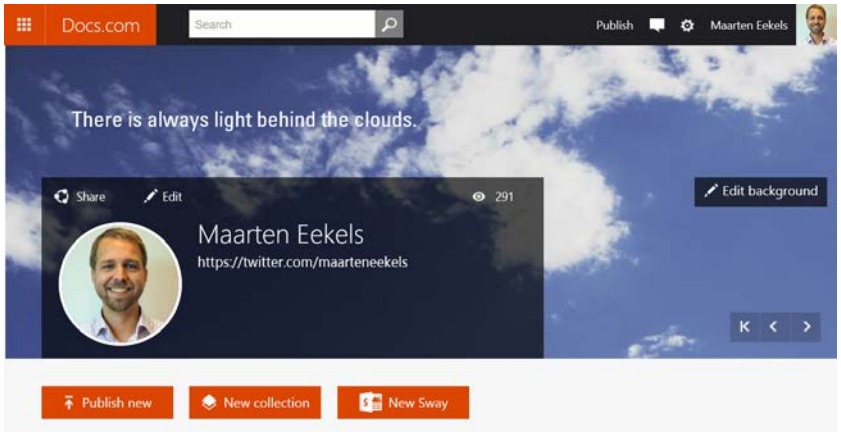
For further details about the PnP Remote Provisioning Engine, you can also read the corresponding documentation published in a dedicated MSDN section (https://msdn.microsoft.com/en-us/library/office/mt604894.aspx).

Conclusion

The goal of this article was to briefly introduce you to PnP, to what PnP does, and to the main projects and topics covered by the PnP project. By reading these lines you learned something about the PnP Core Library and the PnP Remote Provisioning Engine. It is now time to start downloading the NuGet package, as well as the corresponding PowerShell extensions. Play with the provided tools and libraries, and feel free to come back to the community with feedbacks, new ideas, and with your own contribution. PnP is a community project, and you are part of the community.

Docs.com

Docs.com by Microsoft is a recently launched platform where you can publish your Word, Excel or PowerPoint documents, retaining all formatting and animations. Or you can publish new formats like Sway and Office Mix. You can use either your Microsoft Account or Organizational Account (Office 365 Account) to create a profile and start sharing your documents. You can use the platform for free!

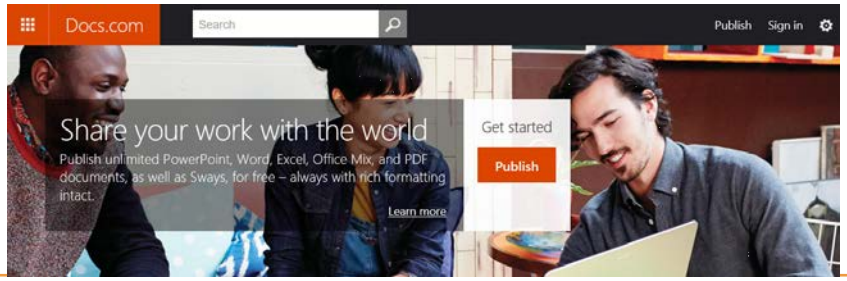


Once you have registered and created your profile, you can start organising your content in collections. Also you can browse collections created by others. You can like content from others, start conversations, or download content. Docs.com makes it easy to share your documents and collections via Facebook, Twitter, e-mail, and more. Or use the embed functionality to embed docs and collections into your blog or website.

Once you have uploaded two or more documents, you are able to personalise your Docs.com URL. Please note, that this is a one-time exercise. Once set, you cannot change it into something else.

Happy sharing!

Maarten Eekels



Information management Office 365

by Albert Hoitingh

What do SharePoint, leakage of information and government agencies have in common? Well, to be honest, not too much. But they did inspire me to write this article. An article about the resignation of a minister, the willful destruction of official documents and the Microsoft Office 365 platform.

Introduction

There has been a little upheaval in Dutch politics recently. A Dutch radio broadcast (http://argos.vpro.nl/nieuws/voormalige-ambtenaar-geeft-toe-vws-vernietigde-documenten-uit-vrees-voor-wob-verzoeken) reported on willful and sanctioned destruction of official government documents. In the broadcast a civil servant admitted destroying these documents and emails in order to circumvent the Dutch Freedom of Information Act or WOB. When first asked about these documents, the departments claimed that these documents or emails did not exist.

Problem was… the reporters of the program had received these "none existing" documents from a whistleblower. So, how come the documents could not be recovered, when they did exist? And how is it possible to destroy vital documents within a governmental environment?

In another example (http://nos.nl/nieuwsuur/artikel/2023736-bonnetje-gevonden-opstelten-en-teeven-stappen-op.html), a bank statement was lost. The bank statement could prove or disprove the allegations against the state secretary for the Ministry of Security and Justice. Unfortunately, according to officials, this bank statement (pertaining to legal proceedings and 1.7 Million Euros) could not be located. To make matters worse, details of the bank statement did find their way to another team of reporters, resulting in the resignation of both the minister and state secretary.

Having worked with Microsoft technology and more recently document- and record management, I find this kind of reporting intriguing. Could this have happened should Microsoft SharePoint been used correctly and effectively? In this article I will try to give you my view on this.

Independent inter-departmental inquiry (https://www.rijksoverheid.nl/documenten/rapporten/2015/12/09/rapport-oosting December 9th 2015):
"… the reconstruction of facts was difficult due to an insufficient organized filing system combined with a lack of proper management…"
Report by the Cultural Heritage Inspectorate\Archive division (http://www.erfgoedinspectie.nl/binaries/erfgoedinspectie/documenten/rapport/2015/12/09/de-waarde-van-archief/de-waarde-van-archief.pdf):
"… the State Inspectorate concludes that although archiving has improved, a complete, organized and accessible state of archiving cannot be fully guaranteed…"

Disclaimers alert

I will never claim to know everything about the subject of document and record management, data loss, physical records or collaboration. To be honest, it's too much to describe in one article (which has been reduced to fit into this DIWUG anyway). If you really want to know all about the examples I use, please download and read the reports as I won't go into much detail.

And also, Office 365 is not perfect either (see this whitepaper http://download.1105media.com/pub/mcp/Files/Mimecast_The_Role_of_Third-Party_Tools_for_Office_365_Compliance.pdf). So, I have to make some disclaimers. I do have my thought on these subjects, so feel free to email me if you like.

In this article I won't go into sustainable storage of records nor will I start any discussion on document formats either. I also won't go into the realm of case-management or project-management. These kinds of functions mostly require specific record management solutions directly related to the function.

I will not discuss functions like OneDrive for Business or the new Office 365 Groups feature. These functions are great and an absolute killer-app for document collaboration and document sharing. Unfortunately, none of the solutions in this article are applicable to these functions. That is to say, at this moment in time.

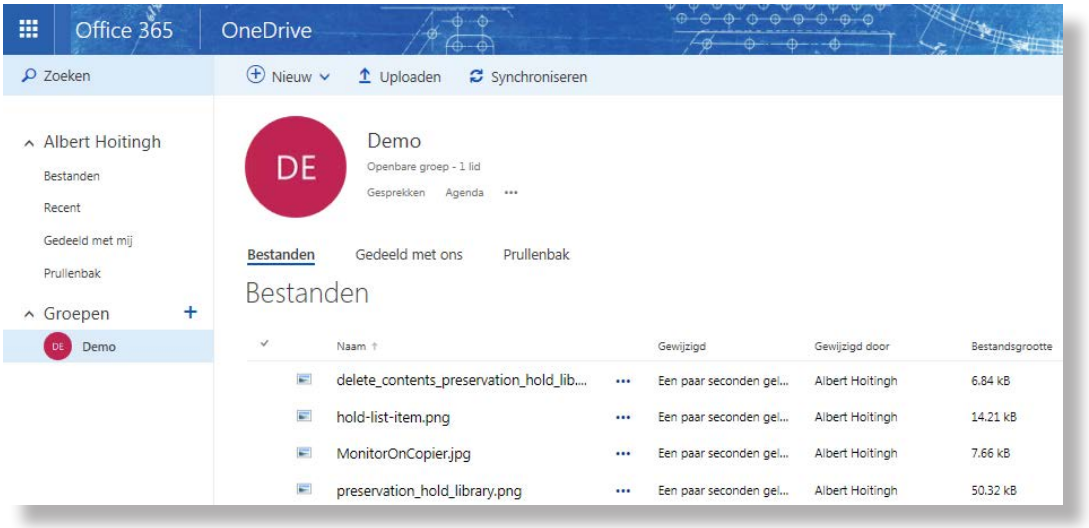


Figure 1: Office 365 groups with Onedrive for Business

Part 1 - Information management

Documents, records and their lifecycle

Documents and emails come and go within an organization. Documents are stored in file cabinets, on file shares, SharePoint, IBM Lotus Notes, another DMS or DropBox. Emails are mostly handled by Microsoft Exchange or IBM Lotus Notes. We use these documents and emails as we see fit. We open them, modify them, forward them to people outside of our organization, and delete them.

In the end, the organization ends up with Terabytes of digital content. Some of this is still relevant. Some of this needs to be retained for a certain amount of time and then deleted. And some if this should never be modified when it enters or leaves the organization.

What?

Not all digital content needs to be retained. Most organizations will have guidelines describing what needs and needn't be retained, this is called a Selectielijst (selection list) in Dutch (https://www.rijksoverheid.nl/binaries/rijksoverheid/documenten/besluiten/2011/06/20/selectielijst-vng/selectielijst-vng-actualisatie-2011-versie-terinzage-bijlage-2-en-3.pdf). Crucial content, like the documents and bank statement in my example would have to be retained.

Why?

When working with content, some of it is used, shared and sometimes deleted. But some content needs to be retained for a certain amount of time. In the Netherlands this is governed by the Archiving Law (Archiefwet). This law requires governmental organizations to maintain records regardless of form (paper, microfilm, digital) as evidence of social accountability.

Part of this law is the NEN-ISO 15489 standard, which defines record management.

In our examples, the governmental departments were bound by this Archiving Law (Archiefwet). Emails and documents are part of this law and should have been recoverable during their retention period (see below). It should never have been possible to manually delete these emails or documents without the knowledge of record managers.

Figure 2: Dutch National Archive

Organizations which are not part of the government can or cannot be bound by this type of law. Most organizations however will base their record management processes on this type of law as it describes what, when and how record management should function.

When?

At what point in time does a document become a (legal) record? According to the standard document lifecycle, the process of document creation and collaboration is just the start. Record management (retention and disposition) is (mostly) at the end.

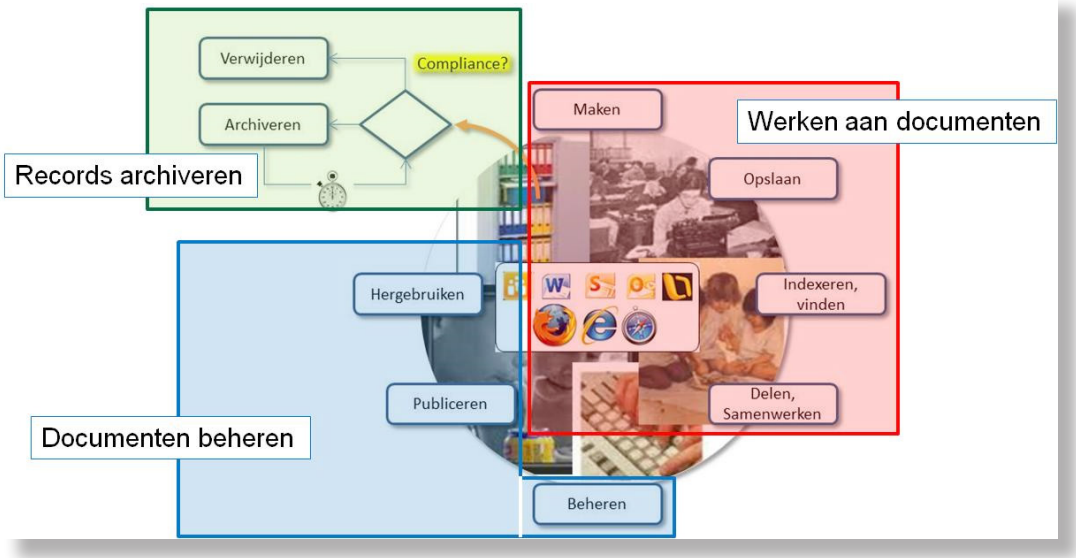


Figure 3: Document lifecycle

(Sorry about the figure. It's in Dutch and has been around for a while.)

At some moment during this lifecycle and if required, a document becomes a record.

The question of "when" this should happen is prone to discussion.

Some organizations take the view that records are created when a certain event (a trigger) occurs. This trigger might be a change of status ("case closed") or when the organization receives an incoming (scanned) document. Other organizations regard all content created during their primary processes or used within the decision making process to be an automatic record.

For how long?

Record management is not just about retaining content for a certain amount of time. It is also about removing this content if allowed or needed to prevent over-retention. There's a fine balance between retaining enough content and over-retention. In most organizations there will be over-retention, simply because our users are not prone to removing content by themselves (remember all those file shares we have?).

It should be clear which content needs to be preserved, for what period of time and what needs to be done after this time. This can depend on the context and/or the type of content. Content created within a project might be suitable for retention, while content within an informal community might not be.

Most organizations have guidelines which states what information needs to be retained and for how long. In Dutch, this is called a Selection List (Selectielijst: https://www.rijksoverheid.nl/binaries/rijksoverheid/documenten/besluiten/2011/06/20/selectielijst-vng/selectielijst-vng-actualisatie-2011-versie-terinzage-bijlage-2-en-3.pdf).

Part 2 - SharePoint

In this part it is time to delve into the more functional and technical side of the equation and look at some of the functions offered by Microsoft SharePoint (with a little bit of Microsoft Exchange or Office as well).

SharePoint Record management

Record management enables you to "freeze" a document for a certain amount of time. The record nor its metadata can be edited or deleted. SharePoint offers in-place record management, record centers and site policies.

In-place records

When using in-place record management, the document stays in the document library and becomes a record. It cannot be changed, the metadata cannot be changed and it cannot be moved by hand. There are two advantages to this. First of all, the document stays within its context. And secondly, the access rights to the document don't change. If needed, a retention scheme can be defined for this record and this will start right away.

Figure 4: SharePoint records

Record centers

An alternative to in-place record management is the record center. The record center is a specific type of site-collection. Documents sent to the record center are stored in record libraries. The retention period is normally placed on the record library. If needed, you can use multiple record centers for different types of records.

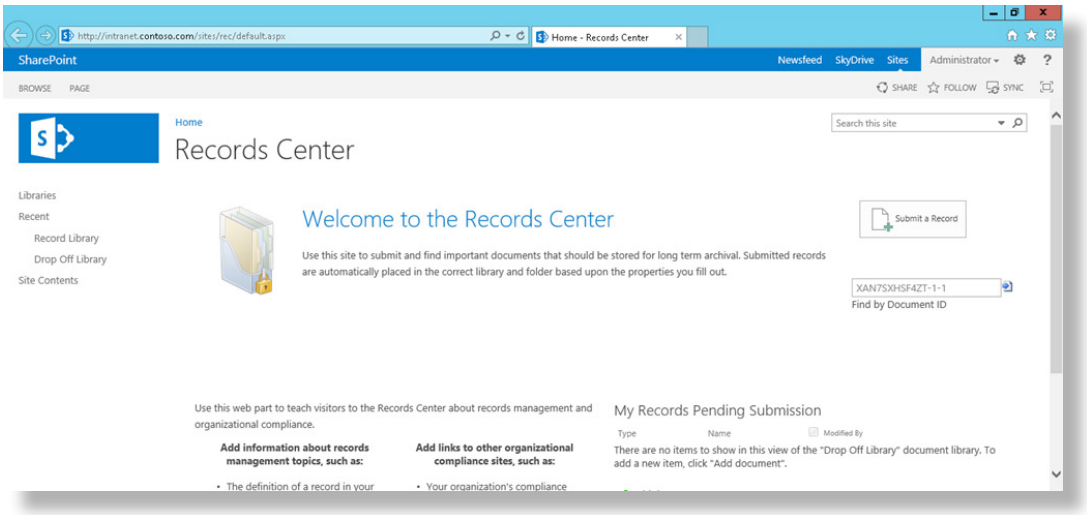


Figure 5: SharePoint Records Center

A combination of both is possible. Use in-place record management first, followed by a move to the record center later.

Site Policies

There are scenarios where it is imperative to store all content within the context it was created. An example is a (long running) project or case. The scope of the content may exceed the scope of documents. It might be newsfeeds or tasks as well. In fact, financial institutions are required by law to archive social media content.

In this kind of scenario, the site policy is a winner. Using the site policy, you create an in-place record of the whole site-collection. When the retention period has expired, the site can be automatically deleted (if policies allow).

Triggers

SharePoint needs some form of trigger to either create a record or send a document to the record center. SharePoint's standard triggers are based on date/time. In an on premise environment you can create your own. One example of such a trigger and action might be:

> ***If a document has not been modified in the last year, create an in-place record. One year after record declaration, move the record to a record center.***

Other triggers can be based on the type of document. For example, you might want to create an in-place record when a document enters or leaves the organization. In case-management scenarios the trigger might be the closure of the case.

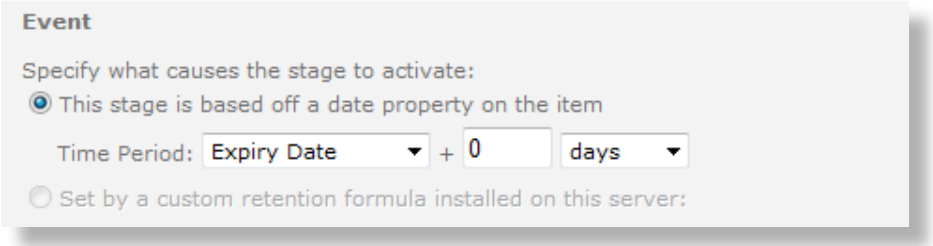


Figure 6: The trigger for record creation.

Please note that although you cannot use a custom retention formula in SharePoint Online, the option still presents itself.

For how long?

We can use SharePoint record management to govern the retention of records. This ensures that records are deleted on time or that record managers are given a heads-up when records need to be moved to an external archive (for instance, the National Archive).

The retention period is based on the date a record is created. Most of the time the organization will have different retention periods for different kind of content. For example:

- Police reports: max 0,5 years
- Correspondence: max 3 years
- Judicial information: Indefinitely (to be transferred).

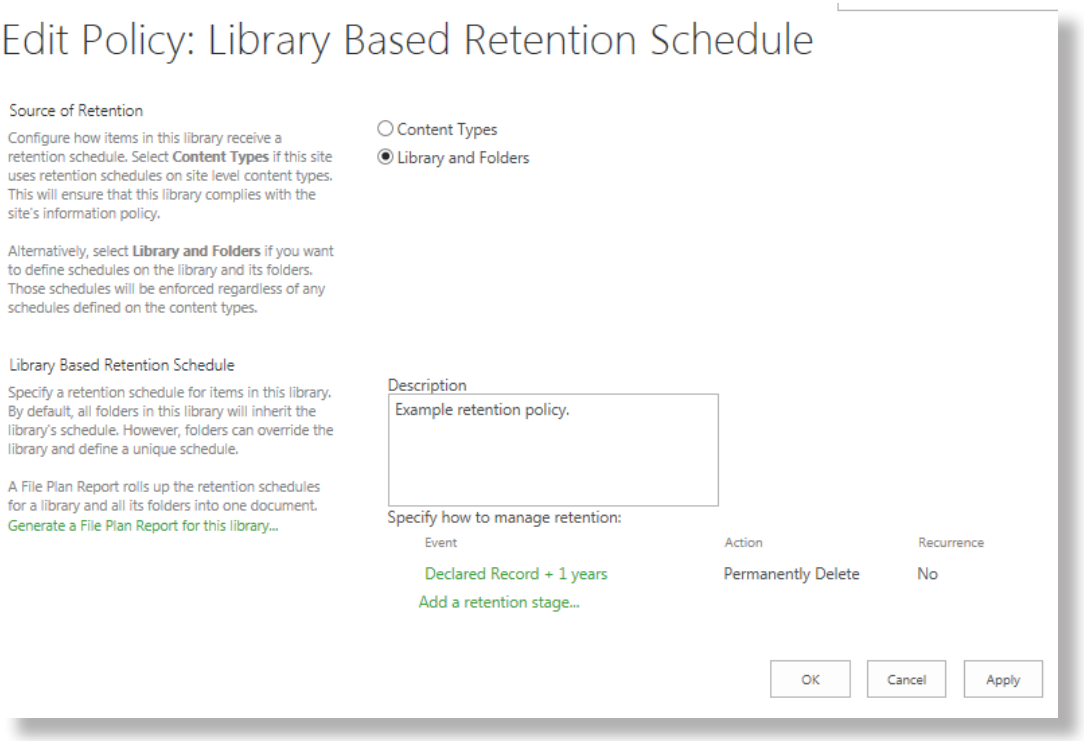


Figure 7: Retention schedule

SharePoint supports this using retention policies. The retention policies work on either content type level or on library level and can have multiple stages. When active on the library level, any policies based on the content type will no longer work.

What about physical records?

In the previous section of this article I have talked about electronic record management. But most organizations still have physical records, or at least a lot of paper. You can use SharePoint to manage this physical archive as well. You might use a standard SharePoint list and add some metadata, for example:

- Business Owner
- Current location (from the termstore)
- Record status (from the termstore)
- Barcode
- Destroyed? (Y/N)

Barcodes can be added to a document from SharePoint when it is printed. When scanned, the SharePoint metadata are displayed.

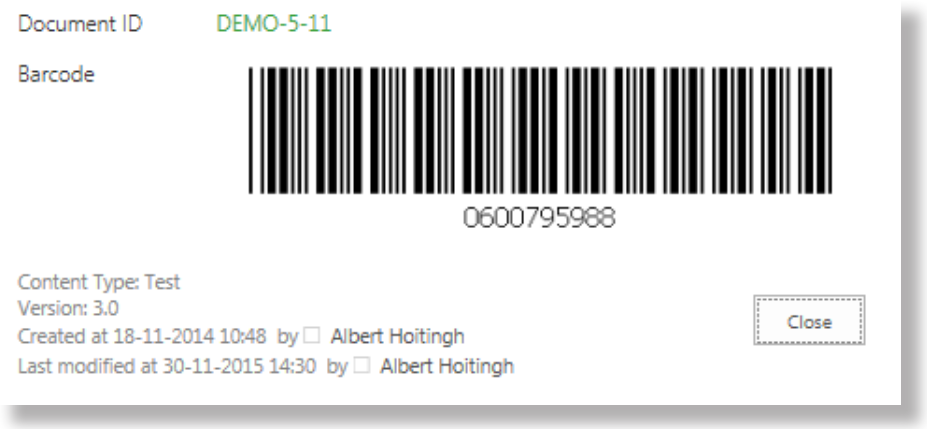


Figure 8: Barcodes and SharePoint

You can use Excel to export the list and print labels for the physical boxes. If you want to track the changes to an item, you can simply use the versioning feature of the list.

Off course, this is a very basic solution using standard SharePoint functions.But a lot of organizations still use Excel (or even Access) to manage their physical archives. SharePoint might be the better option.

SharePoint finding and retrieving

Finding and retrieving the content

Retention of content is only effective when it can be retrieved over time. Remember the example at the beginning? The bank statement could not be found, because no one knew where to look or how to find it. Eventually it ended up in the wrong file.

As luck would have it, information regarding the statement did find its way to a Dutch journalist. It became obvious that the ministry had an ineffective way of finding archived information.

Figure 9: Metadata regarding the bank statement

So, what can SharePoint bring into the mix? Just enabling SharePoint search will not be effective. Correct metadata is needed to enable the retrieval of content. Remember all those file-shares with multilevel directory-structures? And all those documents which are not tagged and can only be “found” using the filename?

With SharePoint this can happen as well when you don't have a correct metadata model and use it wisely. SharePoint offers great functions to support this. The termstore and content type hub, for example. The record management solution needs to provide additional metadata on retention scheme, date of record creation, responsible entity within the organization, record holder, etc.

With content scattered around in teamsites, OneDrives, projectsites, physical archives and record centers it is imperative to enable a form of eDiscovery. The SharePoint eDiscovery template (in Office 365 part of the Compliancy center) offers you a perfect starting point. Use this to discover content, create exports, reports or enable in-place site holds (see below).

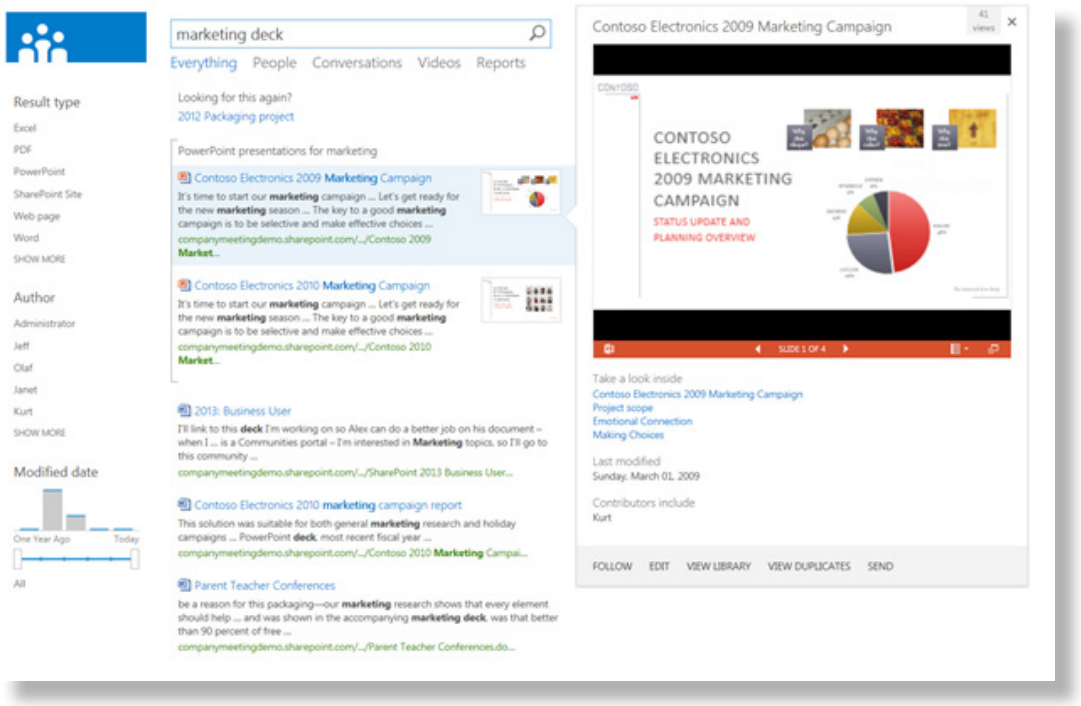


Figure 10: SharePoint search

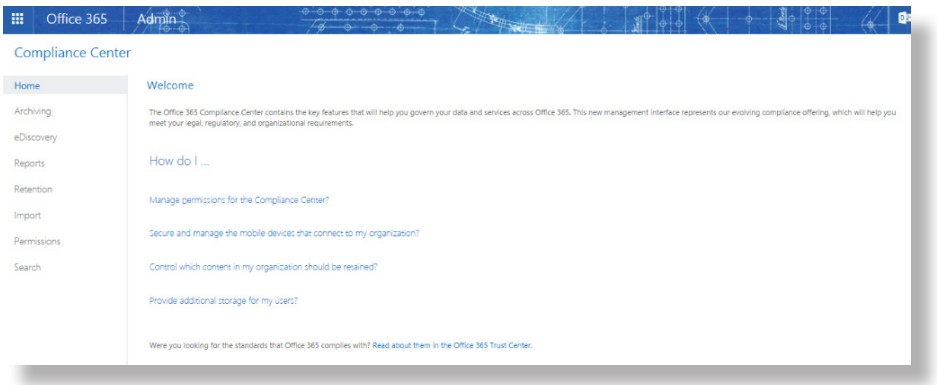
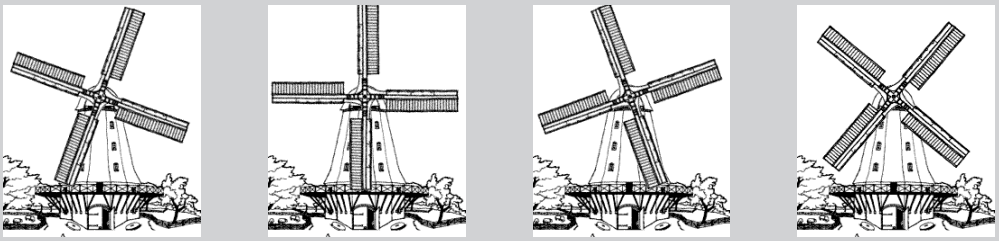


Figure 11: Office 365 Compliance Center, including eDiscovery

The Language of the Dutch Windmills

Joy | Short rest | Mourning | Long rest



Source: Wikipedia.org

Documents deleted during their lifecycle?

When we create and share documents, lots of these documents are inevitably deleted without being a problem. If needed, you can use the (multi-stage) recycle bin to restore documents. But what if you don't want documents to be deleted at all? Here're a few options.

Make a record of all documents.	Usable for documents which enter or leave the organization and should not be modified. Unusable when documents are still in the creating, sharing and collaboration phase. People need to be able to modify their documents.
Just use the recycle bin.	Yep. That's an option. But the recycle bin is recycled itself.
Move documents to a different location when they become final.	A more intelligent way of looking at the document lifecycle. In the next phase of the lifecycle the document is managed differently and, for instance, deletion is made harder.
Use in-place site holds	When SharePoint notices the deletion, copy the version of the document.

In-place site holds

SharePoint 2013 introduced the use of in-place site holds. And these holds might just be the right solution when you want your people to share and collaborate on documents and in the meantime be certain that documents don't "incidentally" disappear from the site.

An in-place hold creates a new (hidden) library called the preservation hold library. Only site collection administrators have access. Whenever a document is modified or deleted, a copy of the latest version is created in this library.

Let's say that someone in the site has deleted the document and has removed it from the recycle bin. With the in-place hold active, this document can still be recovered.

The in-place hold is a compliancy feature. You won't find any option to enable or disable it within the site settings or the site collection setting. Instead, the hold is enabled from the eDiscovery center.

Let's take a look.

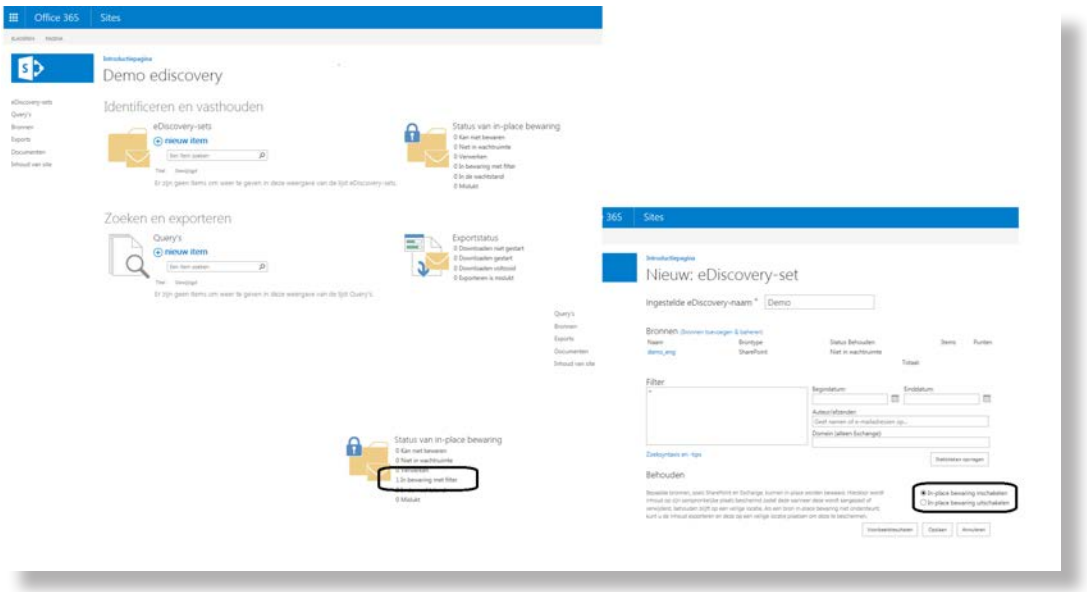


Figure 12: Using eDiscovery to enable an in-place site hold

Figure 13: After the site collection has been processed, a little lock icon is placed behind the URL in central administration. Cool.

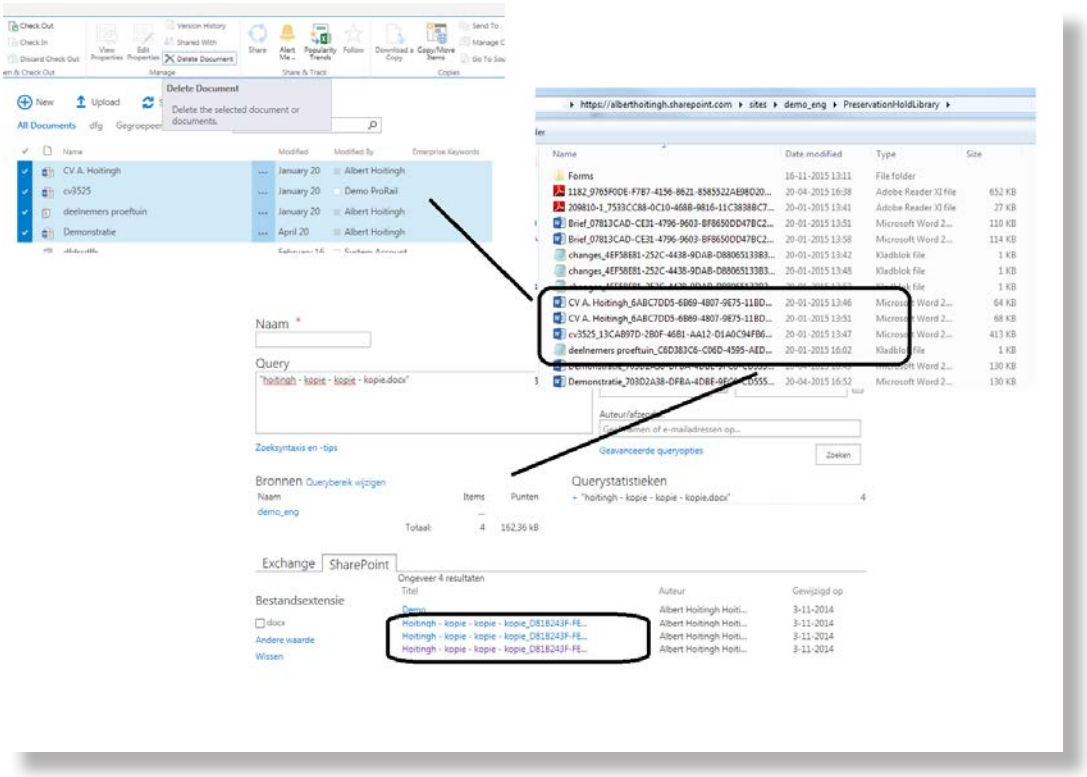


Figure 14: Deleting documents and the preservation hold library

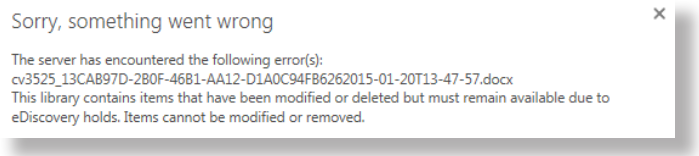


Figure 15: Try deleting documents from the preservation hold

Data loss prevention

Record management ensures that content is retained within the organization for a set period of time. But what if you want to ensure that content does not leave your organization when this should not happen? In that case, you might want to look into data loss prevention. And Office 365 provides several options for this.

Figure 16: Office standard features for document protection

Information Rights Management

Office 365 provides IRM (Information Rights Management) for content stored in the SharePoint environment. IRM enables you to control what people are allowed to do with certain documents. In effect, a document is protected by using encryption technology.

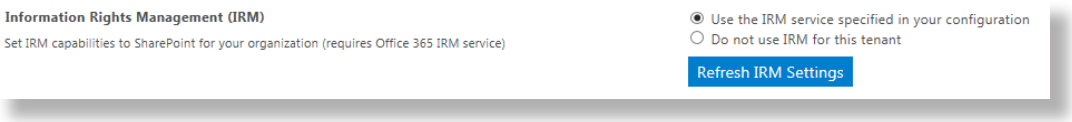


Figure 17: Information Rights Management in Office 365

Using this you can restrict people from printing the document, copying pieces of the document, modifying the document or even making a screenshot of the document. You can even set an expiration date. After this date, the document will no longer be available. Please note though, that even IRM will not stop people from taking a picture of the document or manually copying its content.

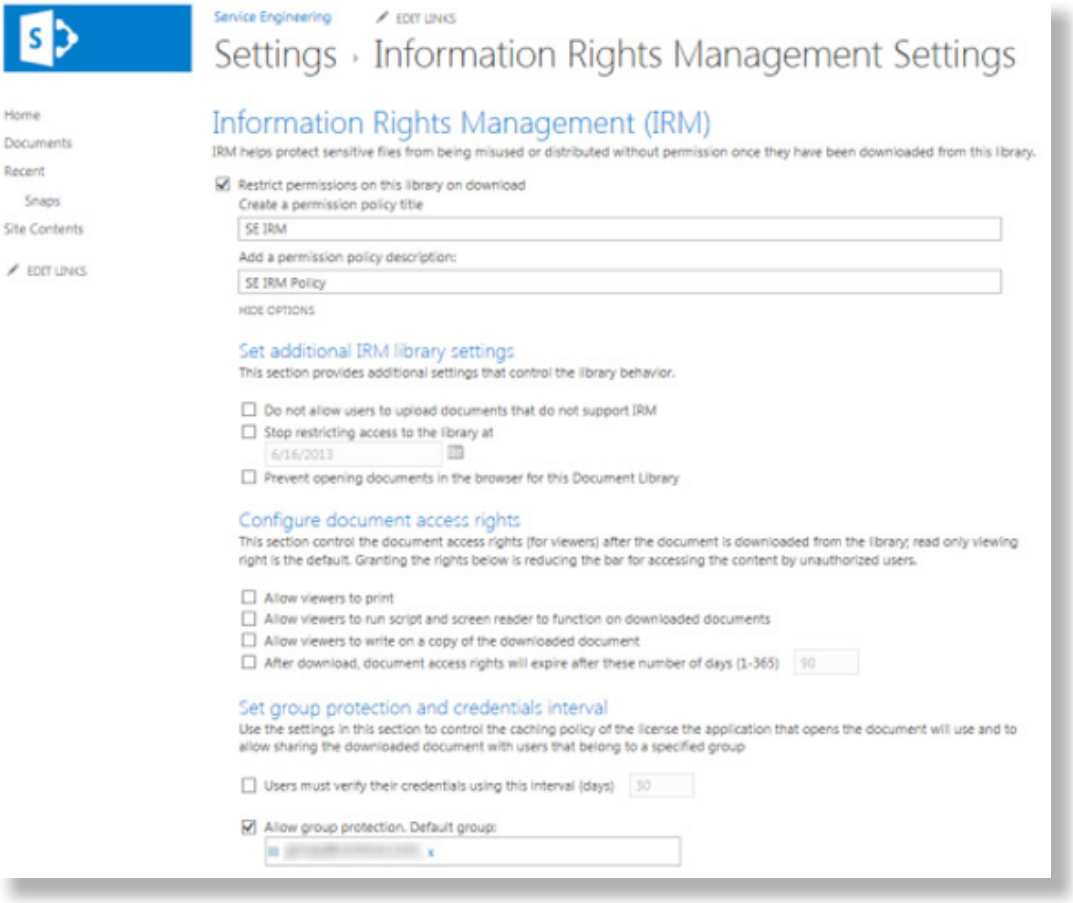


Figure 18: Enable IRM in SharePoint

Use Office

Microsoft Office itself has several options to protect the document. You can encrypt the document or mark it as final. These are very basic, but can be used as a first step for information leakage prevention.

Use Exchange

Exchange Online provides several first rate features for data loss prevention, ranging from policy tips, discovering sensitive information being send and even document fingerprinting. These type of features make it possible to scan email and attachments and determine if there might be sensitive information contained within. (https://technet.microsoft.com/en-us/library/jj150527(v=exchg.150).aspx)

So, it's all about IT then, right?

SharePoint and Office 365 are tools enabling information management on an organizational scale. Any organization with mature information management processes will have some form of record management in-place. But it doesn't stop there. The success of information management and their (mis)use depend on the People, Platform and Processes within the organization.

People: People within the organization need to know how to handle information. Same simple questions need to answered, like is information in an email confidential or should I use OneDrive for storing and sharing confidential documents?

In all, people should be aware of sensitive content and the way this content is preserved in the organization. Inform the people about this. Ensure a culture of sharing but also of compliance and ensure that you have people within the organization specialized in record management and the Freedom of Information Act.

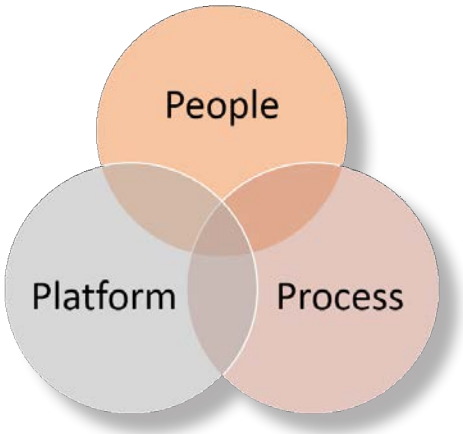


Figure 19: The P3 model

Platform: The platform which support the creation, sharing and publication of content needs to ensure that the content is safe. It should be safe from unauthorized access. The platform should also ensure that the content is protected from unwanted deletion, sharing or modifications.

Process: The organization need to have the processes in place for governing the platform and content. The process for record management should be clear, including the process for record disposition.

The Insider Security Threat, and How SharePoint / Office 365 Measures Up

by Peter Bradley

Of all information security threats, the 'insider' security threat usually isn't the image that people's minds go to first. Angry young men in darkened bedrooms, perhaps. Or offices full of reclusive geniuses, hacking away at the behest of their government, may be.

The costliest form of information security incident is none of these things. It is the incidents caused by regular, every day staff members, acting maliciously, accidentally or obliviously to the damage they could cause by their actions, or lack thereof. Causing or enabling the leak of information to people outside the organisation, or to others inside the organisation who shouldn't have it.

A recent survey (http://www.securonix.com/insider-attacks-were-the-most-costly-breaches-of-2015/) found that security incidents caused by malicious insiders cost companies an average of over $144,000 each year. Another survey (http://www.pwc.co.uk/services/audit-assurance/insights/2015-information-security-breaches-survey.html) found that every company experiences an average of six breaches like this in 2015 alone, and 75% of organisations experienced at least one.

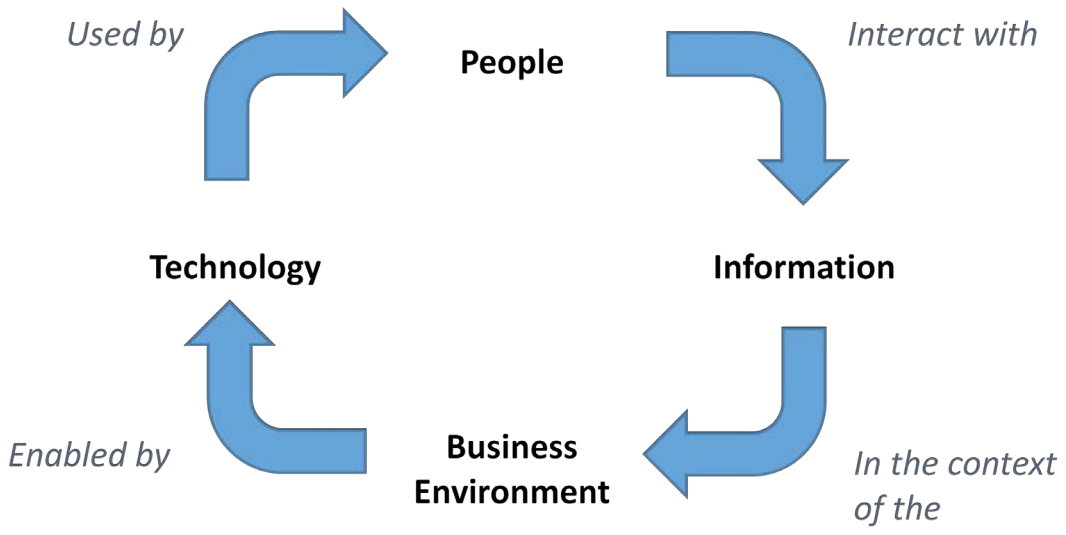
Forthcoming regulations (https://en.wikipedia.org/wiki/General_Data_Protection_Regulation) from the EU are set to apply extremely harsh punitive fines to any company responsible for information containing details of its citizens, which leaks to the public, no matter the cause. Staff member accidentally share a spreadsheet containing customer names and addresses? The fine could exceed €20 million. The regulators - they're not messing around.

This, now, qualifies as a severe, even existential risk to companies everywhere. It is up to us, as practitioners of SharePoint, the biggest information management system in the world, to put information security at the top of our list of concerns in everything we do.

Defining the 'Insider'

For the purposes of this discussion, I'm going to give a definition of the term 'insider'. I'm going to exclude administrators, who often necessarily have high privilege access to information in order to do their job. I'm also going to exclude 'hackers', using subversive technical means to gain access to data.

I'm talking about regular, every day staff members working in the business. And I'm talking about anyone to whom they have given or lost their access credentials.



The Nature of the Insider Security Threat

Most other information security threats stem from gaps in the IT landscape. They tend to warrant predominantly technical responses such as firewalls, DMZs, intrusion detection, etc.

The insider security threat is different. To understand the threat, is to recognise the nature of how information is created, shared, stored and moved within organisations. There are four pillars: people, information, business environment, and technology.

So when it comes to the nature of the insider threat, we can frame it using these four pillars also. Let's list some aspects of the reality of each:

People

- Disgruntled staff
- Phishing threats and social engineering
- People constantly joining, moving, leaving
- External partners / customers / suppliers
- Poor security awareness / too busy to care
- 20% of staff willing to sell their passwords for $1000
  (http://fortune.com/2016/03/30/passwords-sell-poor-sailpoint/)

Information

- Massive and exponentially growing volume of data
- In many different forms, including documents, emails, images, databases, printouts
- Moving independently through different lifecycle stages
- Of various sensitivities, classifications and purposes

Business Environment

- Constant business change - new customers, old partners, departments starting, offices closing, strategies and priorities shifting, organisational structures changing, mergers and acquisitions
- Working practices are increasingly flexible and collaborative
- The best intelligence about the information lies with the people in the business who are closest to the information, but IT tends to be responsible for information security
- IT tends to operate at arm's length from the rest of the business
- Sometimes we try to bridge the gap with workflow and business processes, with varying success

Technology

- Information scattered across multiple platforms of varying maturity and capability
- The rise of the cloud – bringing powerful capabilities more cheaply, but also downsizing of IT teams, skills and budgets
- The rise of mobile devices and workforces – everything is available everywhere, anytime
- Many security solutions and vendors competing for market share

As with any risk, the insider threat can be stated as the likelihood of an incident, crossed with the potential severity of an incident.

So, any effective solution for minimizing insider security threat needs to sit at the junction of all four areas at once, reducing the likelihood and severity of any incident.

Solutions which focus too narrowly on only one or two of these areas are only ever going to have limited overall effectiveness.

Relevant Technologies

There are a wide variety of technology types which are relevant to insider security. Choosing which technologies are right for your business must begin from an analysis of your own specific circumstances, risks and priorities.

Following is a list of some of the relevant technology types. This is certainly not an exhaustive list, but should cover the main areas of interest to most organisations.

- Access Management – controlling and reporting on who has access to what, and how that changes over time
- Data Loss Prevention – detecting, alerting and preventing documents and other forms of information containing sensitive details (such as credit card numbers, DOBs, social security numbers, etc.) when moving across boundaries or between people in violation of business rules
- Behavioural Analytics – monitoring people's interactions with systems and information, detecting, alerting and preventing anomalous behaviour. e.g., if a user tries to download an entire document library when they've not actually been involved with the information, might trigger an alert for an administrator to investigate
- Data Classification – classifying information by some combination of criteria, such as sensitivity, audience or purpose, and then using those classifications to drive security behaviours in both people and technology
- Rights Management – extending and enforcing rights to interact with information in different ways, including when the information has left its host system. e.g. a person may only open a particular document if they have rights to, even after that document was emailed to them outside the network

Other technologies such as authentication, identity management, lockboxes, CASBs, encryption, firewalls, and intrusion detection are also relevant in a general sense, but are more foundational and technical than this discussion is intended to cover.

How SharePoint / Office 365 Measure Up

Some technology types relevant to the insider threat are quite well serviced through SharePoint, in particular Office 365 which is evolving rapidly. Mileage will vary for on-premises deployments of SharePoint, depending on which version of SharePoint you're running, and which related technologies you've deployed alongside it.

- Rights Management – now generally available through Office 365. On-premises, requires Active Directory Rights Management Services (ADRMS).
- Data Loss Prevention - capabilities coming online in Office 365 (https://blogs.office.com/2015/04/21/evolving-data-loss-prevention-in-sharepoint-onlineonedrive-for-business-and-office-applications/)
- Behavioural Analytics – While this is not generally provided through Office 365 yet, the combination of the Audit and Graph APIs is now providing much of the big data that is the basis for behavioural analytics. Expect to see this space maturing over the next 12-18 months.

SharePoint's Biggest Insider Security Gap

In my judgement, the biggest gap in SharePoint in terms of the insider security threat is, and has always been its permissions model.

I'm referring to SharePoint on premises, SharePoint Online and OneDrive for Business collectively here, because their permissions models are effectively identical.

SharePoint's model for managing user permissions tends to be difficult to manage, especially at scale. Configuration errors where people have access to information they shouldn't have, are commonplace. They are also extremely difficult to detect, because doing so requires a close understanding of the people, business and technical perspectives of every individual piece of information (i.e. each of the four pillars above).

A recent survey (http://www.forbes.com/sites/gilpress/2014/12/09/sony-is-not-the-only-company-with-subpar-data-security-new-survey-finds/#46e6d6db2547; not SharePoint specific, but certainly relevant) found that 71% of staff have access to more information than they should. Bringing this number down dramatically is very difficult. For years, Information Architects have prescribed governance processes to address this issue, but success has always been limited.

But minimising this number is fundamental to minimising the likelihood and severity of insider security incidents. If this number were theoretically zero (i.e. every person always has access to only the information they need at any given moment, and none that they don't), this would equate to a minimisation of the insider attack surface, and thus minimisation of the risks.

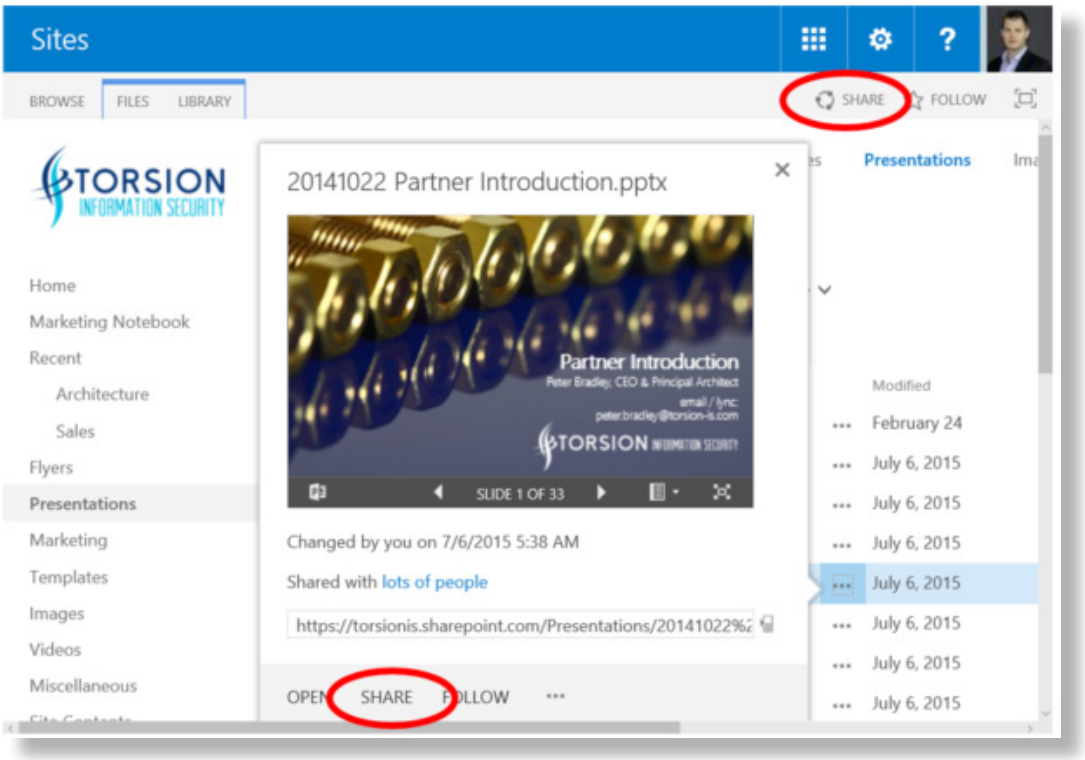
How This Gap Arises – 1. User Experience

Firstly, the SharePoint user experience is oriented towards information sharing – making it easy to grant access for people to information. This is great for facilitating collaboration and information reuse. But the counter-point – revoking access from people to information when it is no longer needed – is much less prominent.

Take the following screenshot of a SharePoint document library as an example. There are two prominent links in the main user experience, suggesting that we expand access rights, circled in red.

Want to revoke access for a person to a document? I count eight clicks (Context menu > … > Advanced > Shared with > Advanced > Select User > Remove User Permissions > OK). And that is only after some trigger for someone to go on this journey in the first place. In reality – its just not going to happen in any normal course of events.

The result is that people tend to accumulate access to information over time. The business circumstances which made it appropriate for a person to have access to a piece of information may have changed, but people keep their access anyway. We call this phenomenon, 'Privilege Creep'.



How This Gap Arises – 2. 'Who' is Not Meaningfully Connected to 'Why'

It is easy to grant access for a person to a piece of information. It is much harder to know why this was done in the first place, unless you happen to be the person who made that that decision. And even then, are you really going to remember why you gave someone access to a document at 2am 18 months ago? Probably not.

How is this relevant? Because unless we know why someone has access to something, we can't easily reassess whether they should continue to have access to something. It is very difficult to determine the correctness of access permissions in the future, especially as the information ages, and people's attention moves on.

And if we can't assess that, we can't revoke people's access when they no longer need it, unless we interrogate the institutional memory of the specific people involved. We certainly can't do that automatically.

But given that we may have thousands of staff, millions of documents, and squillions of individual access control decisions along the way, automatically is the only way it is ever going to work.

SharePoint 2010 did actually provide the basis for improving things in this area, when it enabled role-based access control through claims-based authentication. This allowed us, for example, to provide read permissions for a document to anyone in a given department. This was why someone should have access, not who should have access. It was a great start.

But the unfortunately this functionality didn't fully reach its potential. It required complex customisation to be used in practice. It had scaling issues, was difficult to manage, and most importantly it couldn't be 'just switched on' unless you were a serious SharePoint expert.

And even though this functionality remains in SharePoint 2016, perhaps the fatal blow was that it is not supported in Office 365.

How This Gap Arises – 3. Empowering IT, not the Business

A number of vendors have tried to fill this need, by providing tools which empower IT teams to monitor, manage and audit people's access to information.

But this highlights another fundamental problem – IT tends to operate at arm's length from the rest of the business. Whoever is responsible for controlling who has access to what, and how that changes over time, must be in a good position to understand the context, sensitivity, purpose, and nature of the information itself.

This is only ever going to be the people in the business who are closest to the information. It is never going to be the IT teams – no matter how many business processes we create to try and bridge that gap in understanding.

Conclusion

Information Security is one of the most defining business concerns of modern times. And as practitioners of SharePoint, the biggest information management system in the world, it is our responsibility to put information security at the top of our list of concerns in everything we do.

The insider threat is the most costly type of threat there is http://www.forbes.com/sites/gilpress/2014/12/09/sony-is-not-the-only-company-with-subpar-data-security-new-survey-finds/#46e6d6db2547st costly type of threat there is, and one of the most difficult to address. Because the nature of the threat lies at the junction of the people, information, business environment and technology, any kind of effective solution will require a drastic shift in how we think about the problem.

Just give it a try and contact Marianne, Maarten or Mirjam.

Marianne van Wanrooij
marianne@diwug.nl

Maarten Eekels
maarten@diwug.nl

Mirjam van Olst
mirjam@diwug.nl

A new world of Front-end development

by Cas van Iersel

The way we develop software in the SharePoint space is changing. A Front-end developer wasn't used to having good support and tools in the big IDEs[1] like Visual Studio. But that has changed. Today Front-end development is BIG!

Let's define Front-end development

According to Wikipedia, Front-end development[2] is equal to Client-side development. Basically everything that can be build using just HTML, CSS and JavaScript, and therefor runs on a client's computer, fits into that category. It's even possible to build a complete Client-Server application using just JavaScript. That point taken aside, Front-end development is not just making SharePoint pretty or responsive.

Front-end development and SharePoint

Now more than ever, Front-end development is a skill that is in high demand in the SharePoint Workspace. When SharePoint development is asked for, best practice now is to build most applications client-side. And for a good reason! The App paradigm in general changed our IT world from big full blown applications to a combination of very small pieces of software that focus on doing just one thing very well. Since SharePoint 2013 and the introduction of the App Model, this shift has also occurred in the space of SharePoint Development. The trend was to build full blown applications (Intranets i.e.) in SharePoint, based on deep knowledge of the SharePoint platform and ecosystem. Today this is different. Smaller applications (Add-ins) based on just the knowledge of the SharePoint API's are now the course of action.

Level Up your Front-end skills

Given this new way of how we build solutions, as a Front-end developer an investment must be made to learn new skills. When looking at HTML, CSS and JavaScript these files were written by a developer to be one big file with loads of code in them. One big CSS file or may be a couple to separate styles and same goes for JavaScript file(s). The good news is that today this code can be build, tested and deployed just like server-side code.

CSS? SCSS!

One of these new techniques to write CSS is SASS (Syntactically Awesome Style Sheets) and its new notation SCSS. SCSS is actually a syntax that needs to be compiled by a preprocessor to form a CSS file that can be used in a web-application. The way SCSS is structured makes it ideal to build a stylesheet in a readable and maintainable way! For instance, by using variables that can be reused in all SCSS files and use nesting and mixins to have readable code.

```
$heading-font: Segoe UI, Segoe, sans-serif;
$body-font: Helvetica, sans-serif;
$heading-color: #ff9900;
$primary-color: rgb(0, 172, 233);

body {
 font: $body-font;
 color: #000;
}
```

Listing 1: Using SCSS variables

```
body {
 font: $body-font;
 color: #000;

 h1 {
  font: $heading-font;
  color: $heading-color;
 }
}
```

Listing 2: Using SCSS nesting.

```
@mixin border-radius($radius) {
 -webkit-border-radius: $radius;
 -moz-border-radius: $radius;
 -ms-border-radius: $radius;
 border-radius: $radius;
}

.wp-zone-webpart {
 @include border-radius(10px);
 border:1px solid $primary-color;
}
```

Listing 3: Using SCSS mixins

TypeScript for better JS

Another, very popular, Client-Side language is JavaScript. This year a new version of JavaScript (ECMAScript 6) was released in June after the last version (5.1) dates back to 2009. Just a few browsers support ECMA 6 and none support the full set yet. Basically most of the JS code is still ECMAScript 5 and because JavaScript is not Object Oriented like for instance C#, writing it can turn into a plate of spaghetti very fast.

Fortunately, like with CSS, we can now use pre-processors or compilers to turn supersets of JavaScript into JavaScript. TypeScript is at the moment one of the most popular supersets. TypeScript has some quality properties that helps writing better JavaScript once compiled like:

- Strongly typed
- Excellent Class definitions
- Use of interfaces
- Use of mixins

I mentioned ECMAScript 6 because that version also has some of these properties so it's a good investment to start learning.

```
class Greeter {
 greeting: string;
 constructor(message: string) {
  this.greeting = message;
 }
 greet() {
  return "Hello, " + this.greeting;
 }
}
```

Listing 4: Writing a class in TypeScript example

Cool! So now what?

These new techniques help us to create code partials that are highly readable and very easy to maintain. The next step would be to "build" these code partials to Front-end code that we can use in SharePoint. Therefor we need to Tool up!

Since Nodejs (a platform independent JavaScript runtime) has become insanely popular, lots of tools (especially for Front-end development) has been build based on the NPM (Node Package Manager) system. In this new world of tools, everything is pluggable. This way you can build your very own tailored IDE for every type of project. The best thing about using Nodejs and NPM as a base for this, is that it will run on almost any platform (OSX, Windows, Linux, Docker)!

Writing code in Code

As a true Microsoft fanatic I mostly use Microsoft software, but using Visual Studio as a Front-end developer is like using the biggest Swiss army knife available to just cut some rope. It feels clunky and a little bit inconvenient at some times. Luckily Microsoft understood this quite well and shipped Visual Studio Code[3]. A code editor with a task runner and debug capabilities! Besides VS Code there are some other excellent editors you can use for Front-end development like Atom[4] or Sublime[5].

Set up a folder structure

When starting a code project using Visual Studio Code it's important to create a good folder structure. At root level a distinction between the source files and the distribution (build) files can be made. This keeps the development files and the build files separate. The src (source) folder contains separate folders for "script" and "styles" and every other type of file that can be built using a build task.

Ignore the ".vscode" and "node_modules" folders. Those will be created by Code and NPM.

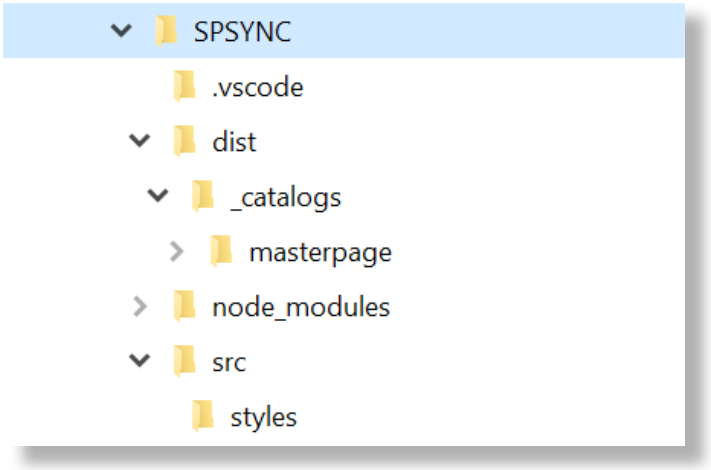


Figure 1: Folder structure

Building code with Gulp

To make this article an easier read, the examples used for building and deploying code are focused on building a custom CSS file. Since now we're working with SCSS we need to pre-process our SCSS files. With Gulp, a task runner based on JavaScript, we can do this in a fast and easy way. Gulp is a good example of a pluggable tool that can be installed in the Nodejs ecosystem. Using NPM we can easily install Gulp into our IDE. Simply fire up a command line (I use cmder) and change the directory to the rootfolder of your code project (SPSYNC). Then type this command to install Gulp.

```
npm install Gulp -g
```

Listing 5: Install Gulp globally

When Gulp is installed a gulpfile.js file is needed in which the Gulp Tasks you want to run are specified. Gulp uses a code-first approach so writing Tasks should feel very familiar.

Let's break down the tasks you would like to perform to build your code and make it ready for deployment. A couple of basic tasks can be:

- Process SCSS to CSS
- Create sourcemapping in CSS
- Minify CSS output
- Copy CSS file to distribution folder

Here's an example build task that does all these things. Notice the variable "styleFolder" which point to a familiar SharePoint folder that we have recreated in the folder structure earlier. This structure is needed when deploying to SharePoint later on.

```
// Load Gulp
var Gulp = require('Gulp');

// Loads in any Gulp plugin
// Attaches them to the global scope
// or an object of your choice.
var gulpLoadPlugins = require('Gulp-load-plugins');
var $ = gulpLoadPlugins();

// Set distribution folder
var styleFolder = 'dist/_catalogs/masterpage/spsync';

// CSS Build Task
Gulp.task('styles', function() {
 return Gulp.src('src/styles/*.scss')
  .pipe($.plumber())
  .pipe($.sourcemaps.init())
  .pipe($.sass.sync({ outputStyle: 'compressed' })
  .on('error', $.sass.logError))
  .pipe($.minifyCss())
  .pipe($.sourcemaps.write())
  .pipe(Gulp.dest(styleFolder));
});
```

Listing 6: Simple build task in gulpfile.js

Another nice thing about Gulp is that not only Gulp itself is pluggable, Gulp uses a same sort of plugin system as NPM. Gulp uses require to load all the Gulp plugins when the Build Tasks runs. Before running this build task, a package.json file needs to be added. This file specifies the dependencies of the plugins we use in this Task.

```
{
  "private": true,
  "engines": {
    "node": ">=0.12.0"
  },
  "dependencies": {
    "Gulp": "^3.9.0",
    "Gulp-load-plugins": "^0.10.0",
    "Gulp-minify-css": "^1.1.1",
    "Gulp-minify-html": "^1.0.0",
    "Gulp-plumber": "^1.0.1",
    "Gulp-sass": "^2.0.0",
    "Gulp-sourcemaps": "^1.5.0"
  }
}
```

Listing 7: Dependencies specified in package.json

Both the gulpfile.js and the package.json need to be in the rootfolder of the code project. When these files are in place "npm install" needs to be typed in de command line (change directory to rootfolder first!) and when confirmed all these plugins will be installed into the IDE.

Deploying code with Gulp

When pre-processing is done it's also possible to do some post-processing and finally use Gulp to deploy files to SharePoint. MasterMind Wictor Wilén wrote a Gulp plugin for this and it works pretty good! To use gulp-spsync[6] first set it up by following the readme.md on Github. Next we add a new task to our gulpfile.

```
// Load Gulp
var Gulp = require('Gulp');
// Load Spsync
var sp = require('gulp-spsync');

// Loads in any Gulp plugin
// Attaches them to the global scope
// or an object of your choice.
var gulpLoadPlugins = require('Gulp-load-plugins');
var $ = gulpLoadPlugins();

// Set distribution folder
var styleFolder = 'dist/_catalogs/masterpage/spsync';

// Settings for deployment
var settings = {
 "client_id":"[YOUR CLIENT ID HERE]",
 "client_secret":"[YOUR CLIENT SECRET HERE]",
 "realm" : "",
 "site" : "https://[YOUR TENANT HERE].sharepoint.com/",
 "verbose": "true"
};

// CSS Build Task
Gulp.task('styles', function() {
 return Gulp.src('src/styles/*.scss')
  .pipe($.plumber())
  .pipe($.sourcemaps.init())
  .pipe($.sass.sync({ outputStyle: 'compressed' })
  .on('error', $.sass.logError))
  .pipe($.minifyCss())
  .pipe($.sourcemaps.write())
  .pipe(Gulp.dest(styleFolder));
});

// Deploy Build Task
Gulp.task('default',['styles'], function() {
 return Gulp.src('dist/**/*.*')
  .pipe(sp(settings))
  .pipe(Gulp.dest('dist'));
});
```

Listing 8: gulpfile.js with build and deploy task

Why use this plugin?

Since this gulp-spsync plugin is in early development stage there are some pros and cons. One of the things that is really cool about this deployment process is that the developers don't actually need permissions in SharePoint. All deployment permissions are covered with an App Principal that needs to be registered in SharePoint first. The con here being that this App Principal needs Full Control on the SharePoint Web and the App Secret needs to be in the gulpfile. Still this approach is neater than adding developer accounts with licenses to an Office 365 tenant and make them admin on a SharePoint web.

Integrate Office UI Fabric in your build

A while back Microsoft released a UI framework to the community. Office UI Fabric can easily be used in Add-ins for Office 365 or SharePoint. Using it will make your Add-in look and feel like a Microsoft product which is great for the end-user's experience. This package can also be installed using NPM and will contain all the source and distribution files. Office UI Fabric comes with its own Gulp builtasks so if you decide to change or improve something you can also build it using the tasks provided.

Copy from node_modules

After installing Office-UI-Fabric the complete package is downloaded into the node_modules folder.

```
npm install office-ui-fabric
```

Listing 9: install Office UI Fabric command

To use it into a code project it's a good practice to move the distribution files into the code project and the distribution folder you use to deploy those files to SharePoint. This will be the next Task that Gulp can perform after building the CSS file! First a good landing spot in the folder structure of the code project is needed. In this example a folder is added to the "masterpage" folder.

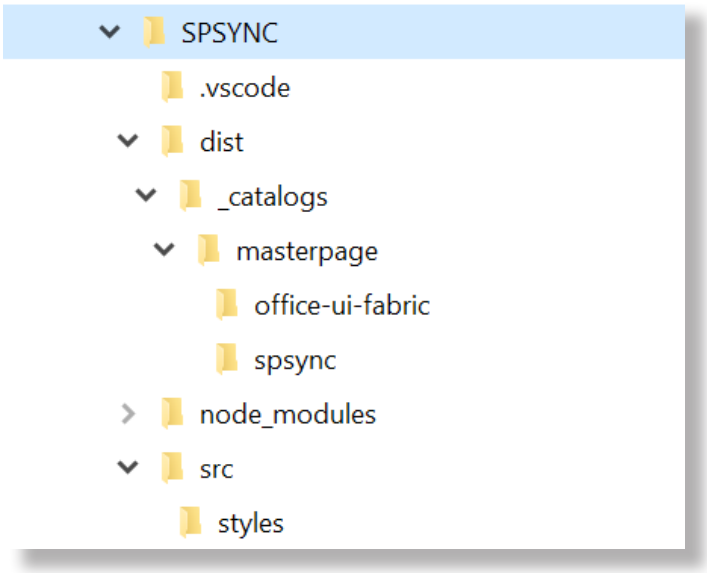


Figure 2: Added folder for Office UI Fabric

Now Gulp need to copy some the JS and CSS files from the node_modules to the distribution folder. This task points to the CSS and JS folder and then combines these two sources into a stream of files. Another plugin is needed for this.

```
npm install merge-stream
```

Listing 10: Install merge-stream

This task needs to get the files from folder "node_modules/office-ui-fabric/dist/js/" and the files from "node_modules/office-ui-fabric/dist/css/" and add them to the folder in our code project we just created. The final task looks like this:

```
// Load Gulp
var Gulp = require('Gulp');
// Load Spsync
var sp = require('gulp-spsync');
// Load Merge
var merge = require('merge-stream');

// Loads in Gulp plugins and attaches them to the global scope
// or an object of your choice.
var gulpLoadPlugins = require('Gulp-load-plugins');
var $ = gulpLoadPlugins();

// Loads SP Sync plugin
var sp = require('gulp-spsync');

// Set distribution folder
var styleFolder = 'dist/_catalogs/masterpage/spsync';
var oufFolder = 'dist/_catalogs/masterpage/office-ui-fabric';
```

```
// Settings for deployment
var settings = {
 “client _ id”:”[YOUR CLIENT ID HERE]”,
 “client _ secret”:”[YOUR CLIENT SECRET HERE]”,
 “realm” : “”,
 “site” : “https://[YOUR TENANT HERE].sharepoint.com/”,
 “verbose”: “true”
};

// CSS Build Task
Gulp.task(‘styles’, function() {
 return Gulp.src(‘src/styles/*.scss’)
  .pipe($.plumber())
  .pipe($.sourcemaps.init())
  .pipe($.sass.sync({ outputStyle: ‘compressed’ })
  .on(‘error’, $.sass.logError))
  .pipe($.minifyCss())
  .pipe($.sourcemaps.write())
  .pipe(Gulp.dest(styleFolder));
});

// Copy Office UI Fabric
Gulp.task(‘fabric’, function() {
 var js =
 Gulp.src(‘node _ modules/office-ui-fabric/dist/js/**.*’)
 .pipe(Gulp.dest(oufFolder));
 var css =
 Gulp.src(‘node _ modules/office-ui-fabric/dist/css/**.*’)
 .pipe(Gulp.dest(oufFolder));
 return merge(js, css);
});

// Deploy to SharePoint
Gulp.task(‘deploy’,[‘styles’, ‘fabric’], function() {
 return Gulp.src(‘dist/**/*.*’)
 .pipe(sp(settings))
 .pipe(Gulp.dest(‘dist’));
});
```

Listing 11: Complete gulpfile.js for build and deploy including office ui fabric

At the end of the gulpfile we see the deployment task which is basically the default task. This task has dependencies on the other to gulptasks “styles” and “fabric” therefor this task only deploys to SharePoint if both other tasks are successfully completed. When this final task is renamed to “default” it will act like the default build task when CTRL+SHIFT+B is hit. But since deploy is a good name it’s also possible to change the tasks.json file that is in the .vscode folder. There change the “taskName” to point to “deploy”.

```
{
 “version”: “0.1.0”,
 “command”: “Gulp”,
 “isShellCommand”: true,
 “tasks”: [
  {
   “taskName”: “deploy”,
   “isBuildCommand”: true,
   “showOutput”: “silent”
  }
 ]
}
```

Listing 12: tasks.json config for VS Code

Conclusion

Hopefully these bits have inspired you to go and fiddle with these new techniques and tooling. The most beautiful part is that this entire scenario can be done on Windows (like I did) but also on OSX for instance! You can literally use the same code and use the same tools! This makes this approach of Front-End Development very flexible. Besides that, only use 54 lines of code are used to support our entire build and deployment process!

If you want to know more about SCSS, ECMAScript6, TypeScript, Nodejs and Gulp here are some great starting points:

http://sass-lang.com for a complete guide to SASS / SCSS
http://www.typescriptlang.org for a complete guide to TypeScript
https://nodejs.org a javascript runtime platform
http://gulpjs.com used for automating our build tasks

References:

1: https://nl.wikipedia.org/wiki/Integrated_development_environment
2: https://en.wikipedia.org/wiki/Front_end_development
3: https://code.visualstudio.com
4: https://atom.io/
5: https://www.sublimetext.com
6: https://github.com/wictorwilen/gulp-spsync

Office 365 collaboration tools: what to use for what?

by Frédérique Harmsze

Office 365 is an extensive toolkit that helps us to collaborate. But there are so many tools in the box and there are so many changes, that users get confused. To collaborate today, should I use OneDrive for Business or Office 365 Groups or Sites or Yammer or what…?

I know I am not the only person who wonders what to use when. Recently, I encountered this question in several organizations that were getting started with Office 365. Some users were annoyed by the overdose of options. Others were enthusiastically starting to use a tool in ways that made me think "oops".

One reason why this question is so acute is that options appear and change all the time, as Office 365 evolves continuously. Another reason is that Microsoft offers a wide range of tools, to suit the needs and preferences of different users in different situations. So there isn't one single tool that you should always use for all collaboration. Still, we can give some guidelines: hammering a nail into a wall using a screwdriver is a valid option, but the hammer is better suited for the job. However, keep the screwdriver at hand too, because you'll need to tighten some screws next.

> ***"Options appear and change all the time, as Office 365 evolves continuously."***

So let's take a look at some of the collaboration tools in our Office 365 toolkit. What are they best suited for and what are they less suited for, considering their strengths and restrictions at this time? We focus on OneDrive for Business, Office 365 Groups, SharePoint Sites and Yammer, because those are the most obvious candidates for collaborating around documents. To avoid our own overload, we will not discuss other tools, like Delve, which helps us find relevant documents, and Skype for Business to talk about them. Outlook, the mother of all digital collaboration tools, will only appear in the discussion of OneDrive for Business and Office 365 Groups, because don't want to collaborate on documents via Outlook by itself.

OneDrive for Business: Your digital desk drawer

Let us start with OneDrive for Business, because several organizations wanted to start by promoting that tool: it offers 1TB of storage space that is already paid for in their Office 365 licenses. OneDrive for Business offers personal storage: the successor of the SharePoint MySite. It also includes a mechanism for synchronizing libraries to your computer, and an entry point for all documents created by me or shared with me anywhere in Office 365.

Use OneDrive for Business for:

- **Storing work-related documents that are relevant only for you.**
  For example, notes about your personal development. Your documents are safely stored in the cloud, including older versions, even when your computer crashes. You can access them anytime and anywhere via the internet, and the synchronization mechanism allows you to work offline too. It is easy to store your documents in OneDrive for Business and then view and edit them in Office: in the client on your computer, in the browser and on your mobile device.

- **Sharing a document ad hoc with colleagues.**
  For example, I stored this article in my OneDrive for Business and shared it with selected colleagues for feedback. That is easy: share the document from the library, or add it as an attachment to an Outlook 2016 mail and choose to upload it into OneDrive for Business. And because it is integrated in Office 365, you can find documents that are shared with you via the central search and Delve. You don't have to open yet another tool.
- **Sharing a document ad hoc with external people.**
  Proposals and other documents may be too big for email. If you do not have an extranet Site for these external people, you can use your OneDrive for Business to give them the document. It is easy to share with somebody outside your organization, if they have a Microsoft account. And you send a guest link to people who do not have a Microsoft account.

Do not use OneDrive for Business for:

- **Systematic collaboration, over a longer period of time, with more people.**
  You are the only owner of the content in your OneDrive for Business. If you leave the organization, your documents are no longer managed. Also, you need to keep track of what you share with whom; you won't see it at a glance. And if you get new team mates, you would need to share all of the relevant files with them too.
- **Storing content that is important for the organization.**
  OneDrive lacks the Enterprise Content Management options that SharePoint Sites have. But what bothers me most is that the content has no owner if you leave.

Some users I talked to were eager to dump all of their documents into their OneDrive for Business, because that is the tool that they saw first in Office 365. Oops… OneDrive for Business is definitely not the best tool for systematic collaboration or content management. So it is not the best place to start when you roll out Office 365 gradually.

> ***You are the only owner of the content in your OneDrive for Business***

By the way, we are talking about OneDrive for Business, not the consumer version, OneDrive Personal. Unfortunately, users often get confused between the two. I have seen users accidentally save work-related documents from MS Word to their private OneDrive.

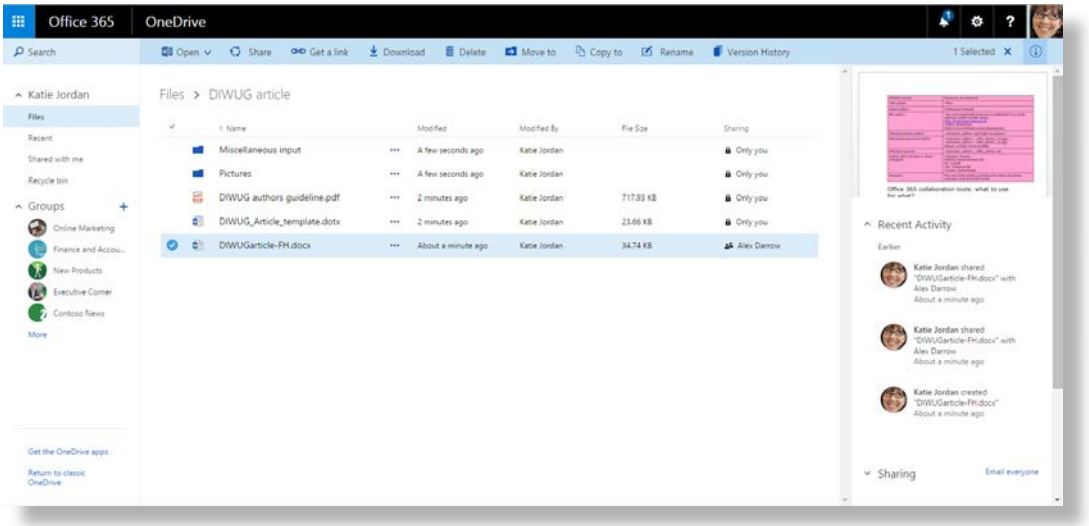


Figure 1: My files in OneDrive for Business, sharing with a colleague to get feedback

Office 365 Groups: Our digital room

Office 365 Groups are relatively new in the Office 365 toolkit, bringing together several existing and new Office 365 services for collaboration. They contain conversations and a calendar from Outlook, with connectors to include posts from other services, like Twitter. You share files in a document library that looks like OneDrive for Business, although it technically is a SharePoint Site Collection. For informal notes, the Group has a OneNote notebook. And now each Group is associated with a plan in the new Office 365 Planner. Also, the Groups appear in CRM and Power BI. So Office 365 Groups are everywhere, including on your mobile devices.

Office 365 Groups bring together several Office 365 services for collaboration

Microsoft emphasizes Office 365 Groups as the future of collaboration. And yes, groups have improved a lot since they were launched last year, although they are not fully mature yet.

Use Office 365 Groups for:

- **Temporary collaboration, for example small projects.**
  Everybody can create an Office 365 Group in a few self-service clicks. You don't have to worry about advanced configuration; there isn't any. You get the standard package and start right away. Microsoft is developing tools to govern the plethora of Groups that this may lead to, like policies to expire inactive Groups and naming conventions.
- **Basic collaboration in general.**
  In the Group, you share everything with its Members, so that new team members get up to speed quickly when they join. Invite a colleague as a co-owner, so that the Group will still be managed when you leave. This is particularly important if you plan for ongoing collaboration, and a key reason to prefer a Group over OneDrive for Business for team collaboration.
- **A shared calendar.**
  If you don't need advanced options but do need a good calendar, try a Group. Its calendar is better than the one available in SharePoint Sites, as it integrates with the members' personal calendars in Outlook.
- **Collaboration with Outlook devotees.**
  Office 365 Groups offer more systematic and richer collaboration than basic Outlook, while still allowing you to live in Outlook. You follow the conversation from your own Inbox, mail to the Group, and open the Group calendar, even in Outlook 2013. But to experience the Office 365 Groups fully, you need Outlook 2016. There you can open all aspects of the Group from the ribbon (see Figure 2). This helps Outlook-minded users to share information in other ways than sending attachments back and forth. For one thing, the attachment is stored in the Files section of the Group…

Ten years of SharePoint knowledge still available at www.diwug.nl

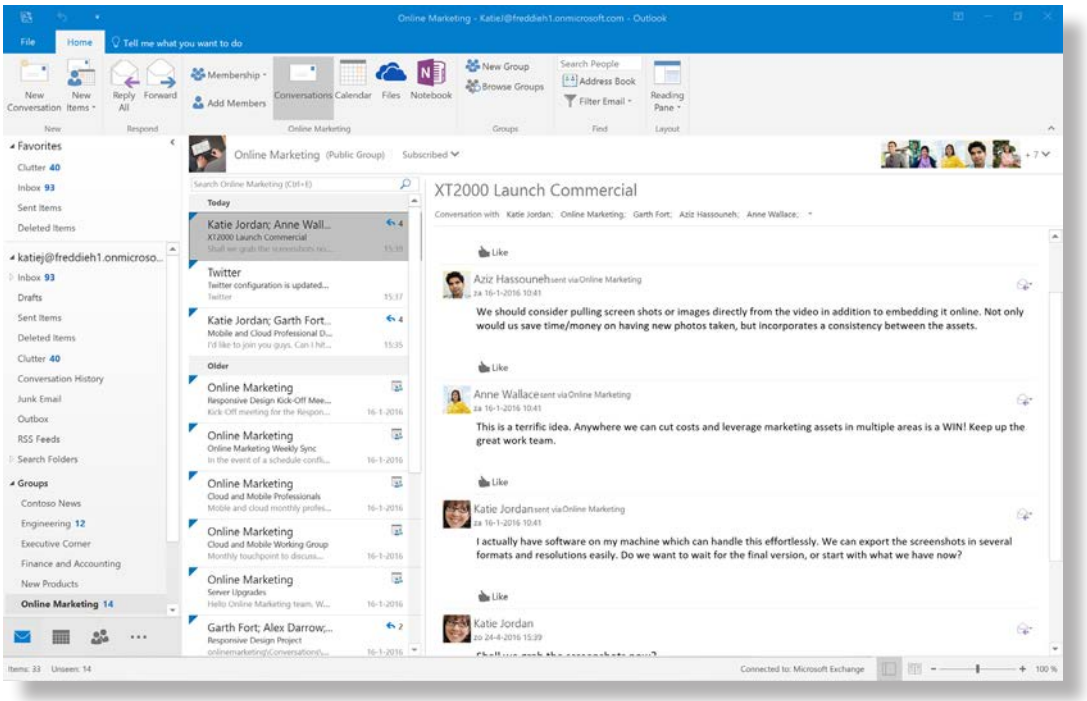


Figure 2: Office 365 Groups in Outlook, starting with the conversation. Buttons in the ribbon open the Group's Files and Notebook in the browser

At this time, do not use Office 365 Groups for:

- **Collaboration without helping inexperience users,**
  especially if they don't have Outlook 2016. Groups are supposed to be plug & play, but in real life, many users struggle. They are not sure how get to the Groups. And they are confused about the switch between the Conversations and Calendar that live in Outlook, and the Files that live in an area called OneDrive. They look different, and there is no overview page that brings them all together. I hope the interface will become clearer and until then: spare your innocent users.

- **Collaboration with external people.**
  People outside the organization can send email to the Group conversation, if you activate it. But for now, externals can't be full Members with access to the files and other elements of the Group. Microsoft is working on guest access support. Until then, the other Office 365 tools are better options for this purpose.

- **Regulated content.**
  The advanced Enterprise Content Management features, like content auditing and site policies, are not available in Office 365 Groups. Some compliance features are emerging though: you can do eDiscovery and litigation holds on Groups, and audit Group events from the Azure Management Portal. If you use a group for "secret" work, be careful: everyone can see the Group title, even for private Groups. Another problem is that all Group content is irrevocably lost, if somebody accidentally deletes the Group; Microsoft is working on an option to undelete the Group. So for now, SharePoint Sites work better for this purpose.

- **Systematic collaboration with advanced processes**
  ith Office 365 Planner, we can assign tasks and keep track of their status in the context of the Group. Unfortunately, there is no link from the Group to its plan yet, as it is still in preview. But it's a start for lightweight project management. However, if you have more or more advanced processes, Office 365 Groups are not the best tool; SharePoint Sites may remain a better option for this purpose

- **Publishing information for many readers, like HR information.**
  Office 365 Groups are meant for collaboration, so all Members can add, edit and delete the information. Also, the information in the Group is not displayed in the most user-friendly way. SharePoint is a better option to publish documents, with an announcement Yammer to talk about them. This may seem obvious, as publishing information for many readers is not collaboration. But some users talked about using an Office 365 Group for that purpose. Not practical.

- **Bypassing the SharePoint Online storage limits.**
  I heard some IT-people wanted to use Office 365 Groups to save their storage quota for SharePoint Sites. Nice try, but that won't work. The Files sections of Groups are implemented as SharePoint site collections, and they are included in the number of site collections and storage calculations.

In any case, the development of Office 365 Groups is very much in progress. By the time you start using Office 365 Groups, some limitations are already resolved. For example, today I saw new options to access the library settings in a Group, create views and change permissions. So a Group is a digital room where you talk with your team and work on documents. The painters and plumbers haven't finished yet, but you can already move in and start using it anyway.

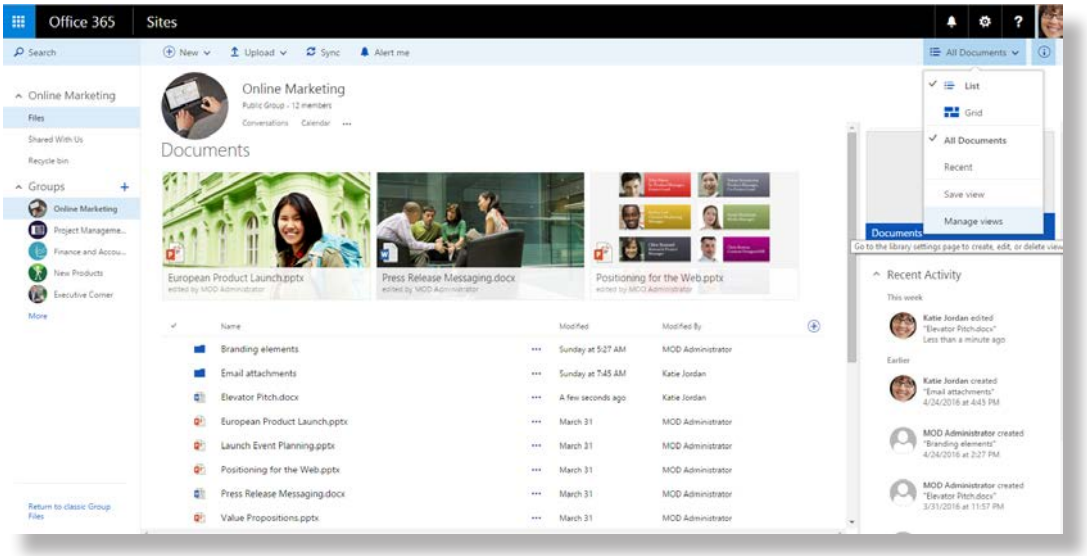


Figure 3: The Files section in a Group, in a view displaying the most recent items and three selected files pinned to the top

SharePoint Sites: Our digital workshop

A SharePoint Site, such as a Team Site or a Project Site, is not a simple digital room. It is a fully equipped digital workshop. It is not the best place for conversations, but if you want to do advanced work or create a great shop window, you have the power tools at your fingertips. SharePoint and its Sites have been around for many years as the key collaboration tool. In Office 365 we have SharePoint Online, though end-users don't see the name SharePoint: only Sites.

Some people have declared SharePoint Online Sites dead: Office 365 Groups are replacing Sites. In my opinion, Sites are not dead yet, and Groups do not replace them for everything.

Sites are not dead yet
and Office 365 Groups do not replace them for everything

Use SharePoint Sites for:

- **Collaboration with processes.**
  In Sites you can set up workflows to facilitate your processes. Even if you don't use full-blown workflows, Sites are better suited for complex processes. For example, with lists where you assign items to people and update their status. In a Site, it is easy to configure such lists to meet your needs. Users get a notification when an issue is assigned to them and they can set alerts on other changes.
- **Collaboration with external people**
  SharePoint Online allows you to collaborate with people outside your organization. If external sharing is enabled by the administrators, external team members collaborate in the Site on the same footing as internal team members.
- **Serious content management and regulated content,**
  for example dossiers that a financial authority has to audit. The Enterprise Content Management functionality of SharePoint allows you to structure the content using metadata and content types like Document Sets, to pinpoint it with unique IDs, and to secure records. The auditing options in the site collection provide you with detailed reports. Fine-grained permissions allow the Site Owner to configure who can see or do what. Sites are invisible to people who do not have permission to access them, so you can do your confidential work in private.
- **Making information available to large groups,**
  such as HR information for all employees. The powerful options of SharePoint allow the Site Owner to configure a Site to be easy-to-use for visitors. Usually, this is a Publishing Site in a portal, as this is not really collaboration, but you can also use a Team Site template. The most relevant information is displayed on helpful pages, which may include smart views on libraries and lists, videos and links to information stored elsewhere. It is possible to allow Visitors access with read-only permission, and to restrict lists or libraries to insiders.

Do not use SharePoint Sites for:

- **Quick & dirty, temporary collaboration with colleagues**
  It takes more time to set up a SharePoint Site than a Group. And if you don't need advanced functionality, a Group is more suited as a throwaway "digital meeting room". Especially if you do not have self-service site creation for Sites. And especially if you do have Outlook 2016, to make Groups easier to use. OneDrive for Business is only for the quickest, dirtiest and most temporary collaboration.
- **Personal documents that are only relevant for you**
  Files that are not relevant for the organization belong in OneDrive for Business.
- **Conversations**
  SharePoint has a list template for Discussions, but that is only good enough for basic conversations in the context of the Site. Office 365 Groups and Yammer work better for conversations; the Yammer conversation can be embedded in the Site.

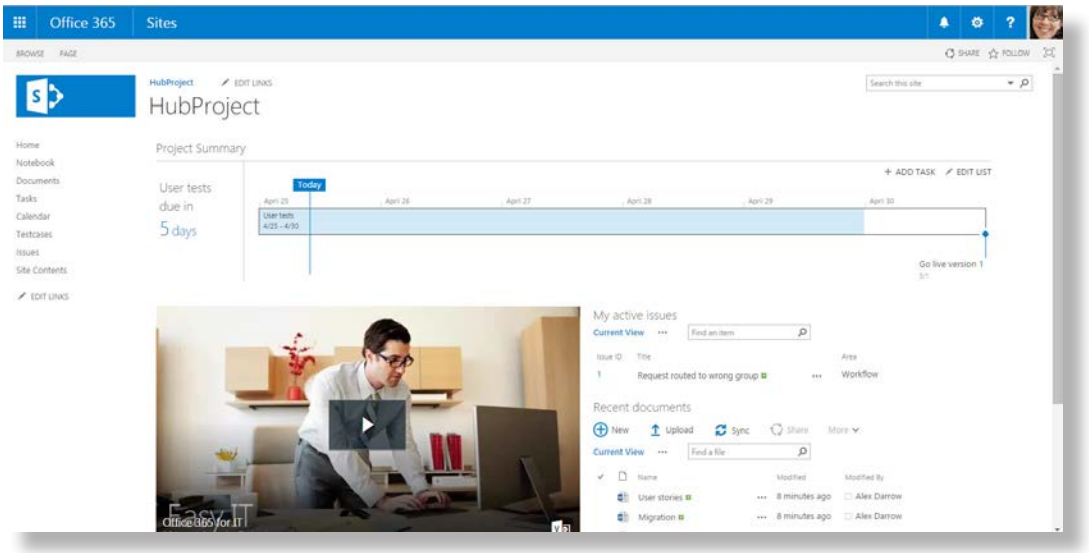


Figure 4: A SharePoint Site displays the most important information on the homepage, such as milestones, active issues assigned to me and recent documents

OneDrive for Business, Office 365 Groups and SharePoint Sites have a lot of useful functionality in common. For example, multiple authors edit documents in the library where they are stored, and then view and restore older versions of the documents. In each of these tools you have a OneNote Notebook which is great for informal notes, gathering information and sharing it. Yammer is the odd man out in our set of collaboration tools, as it does not have any of this functionality.

Yammer: Our digital water cooler

Grab a drink at the water cooler and tell your colleagues what's on your mind. Okay, Yammer does not serve drinks, but it is a great place for informal get-togethers. Yammer is an Enterprise Social Network: it is not about documents, but all about conversations. In private, public or the all-encompassing All Company groups. Yammer was bought by Microsoft and still is not fully integrated with Office 365. But the integration is getting better, now that Yammer lives in the Microsoft datacenter. For example, attachments to Yammer posts show up in your Office 365 Delve overview.

> Yammer is not about documents, but all about conversations

Use Yammer for:

- **Reaching out to the entire organization**
  Yammer allows you to make a formal announcement or post an informal message in the All Company group, so that everybody can read it and respond. Office 365 Groups do not have such an All Company option for global conversation. You could publish the news on the homepage of your SharePoint Portal, but then the employees cannot respond. So Yammer is the best tool for this purpose, especially when you embed it in your portal for greater visibility
- **Discussions and questions in your organization.**
  If you have a question and you are not sure who can answer it, post it in Yammer. I have seen this work to great effect in real life, particularly in specialized Yammer groups. It is easy to post a quick blurb in Yammer and invite people to participate in the conversation. It is also easy to respond and follow a thread. Such groups can have several Admins, so that they stay operational when you leave.

- **Discussing videos or documents stored elsewhere in Office 365.**
  When you add comments to a video in the Office 365 Video Portal, the discussion takes place in the selected Yammer group. But you can view it in the context of the video as well. In a SharePoint Site, you can post a document to Yammer: the document stays in the Site and the discussion takes place in the selected Yammer group. Unfortunately, the option to view the Yammer discussion in the context of the document in the Site has been deprecated.
- **Discussions and questions with externals.**
  Yammer offers a forum for discussions with people outside your organization, in external networks and external groups. There, the external partners can join in all conversations. Be aware that if you link to a document that is stored in an internal Site, the external participants cannot open it. Still, the experience runs smoother than in an Office 365 Group Conversation.

Do not use Yammer for:

- **Systematic collaboration involving documents or processes.**
  A document in Yammer is like an attachment to the conversation: you cannot manage it properly. Both posts and attachments are hard to find in Yammer itself, as its search is limited, and even harder to find from the rest of Office 365, because results from Yammer are not integrated in the general search result yet. So if you want to discuss an important document you'd better store the document in a SharePoint Site and discuss it in Yammer. OneDrive for Business and Office 365 Groups do not offer the option to post a document to Yammer at this time.
- **Posting long stories.**
  Yammer post are hard to read, because you don't have the option to format text. It works better if you publish the long story elsewhere, in Yammer Notes or your Office 365 blog for example, and point to it within Yammer to start a discussion.

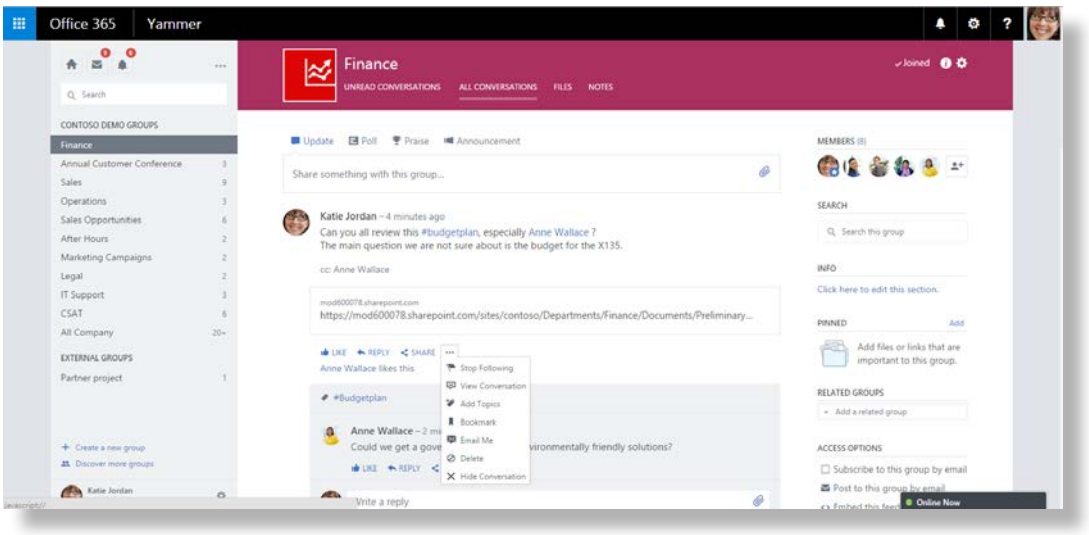


Figure 5: Discussing a document in a Yammer group; the document is stored in a SharePoint Site

Conclusion

Office 365 offers several tools for collaboration, each with its own focus. For each activity, pick the tool that helps you reach your goals. OneDrive for Business provides personal file storage and ad hoc collaboration in documents. Office 365 Groups enable basic collaboration and offer the easiest way to involve users who are wedded to Outlook. Use SharePoint Sites for advanced collaboration facilitating processes, complying with regulations, and broadcasting information to large groups. And Yammer works best for internal and external conversations, including messages to all employees.

This may change as Office 365 evolves. Office 365 Groups will probably become more important when their functionality and usability is beefed up. So the champions of Office 365 should keep monitoring the latest and greatest in the Office 365 toolkit, keep checking what the users need, and keep helping the users take full advantage of the toolkit, to get their jobs done. Not just at the individual level: teams need to align their tools to be effective. And the guidelines for using these tools should fit with the organizations governance. So let's aim for more than just a one-time checklist of what to use for what.

For each activity, pick the tool that helps you reach your goals

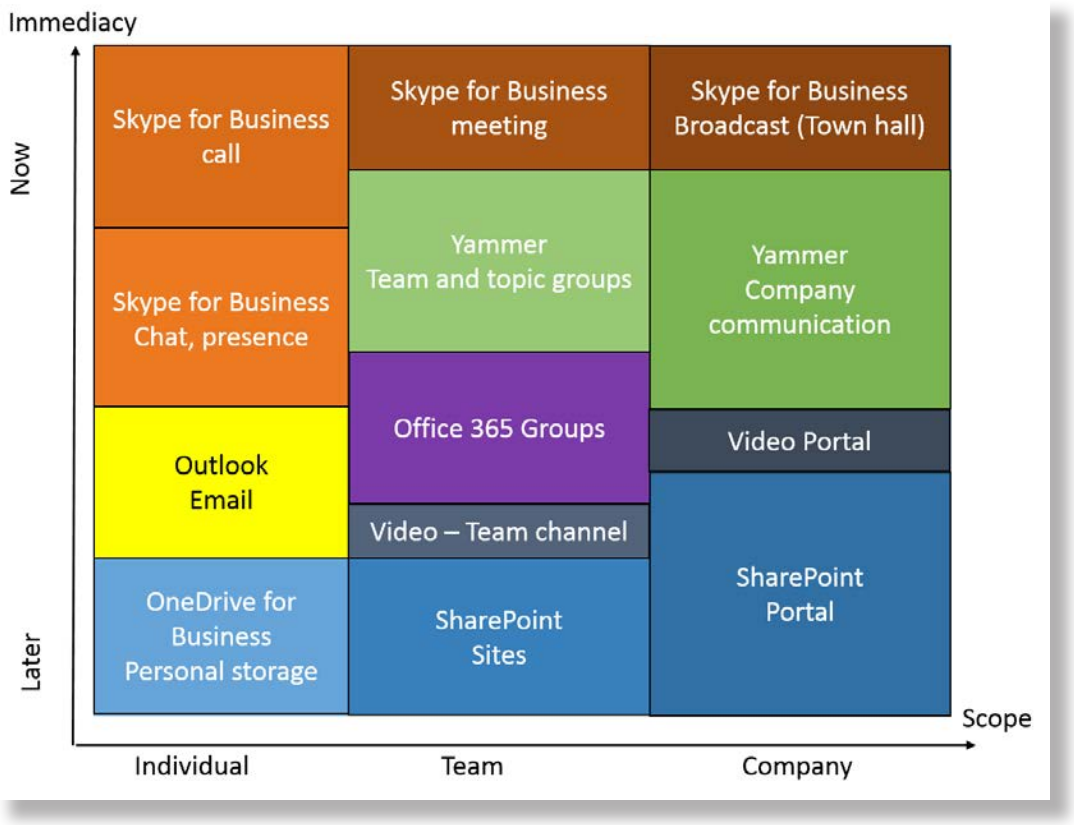


Figure 6: What to use when, inspired by the Harbridge and Khipple whitepaper

References

Microsoft roadmap: http://roadmap.office.com

Richard Harbridge & Kanwal Khipple "When to use what" in Office 365: http://www.2tolead.com/whitepaper-when-to-use-what-in-office-365/

Practical guidance for deploying high trust add-ins on-premises

by Andries den Haan

Developing add-ins for SharePoint is a beautiful thing. Within the projects I'm involved in, add-ins are pretty much standard nowadays.

Although it was an acquired taste for me, I have definitely warmed up to the concepts in the last year. In particular, through the guidance from the Office Development Patterns & Practices (OfficeDevPnP) program.

But here's the thing! There's plenty of guidance on the web to properly deploy remote provider add-ins to Azure and work with them in Office 365, but for on-premises environments the guidance seems a bit limited on some important points:

- Server requirements for hosting remote provider add-ins;
- How to properly set up high trust for a production scenario;
- Requirements for application pools to make a high trust setup work;
- Configuration automation using Windows PowerShell.

The objective of this article is not to go in-depth on all aspects, but to highlight some pitfalls to have a secure and production-ready setup. Some common experience of working with high trust add-ins and web deploy packages is assumed as these topics are not covered in detail.

Setting the scene

I have come to believe in the use of the add-in model, remote provisioning patterns and the basic principle of platform hygiene. Keeping your SharePoint servers clean and avoiding running custom code on any of the boxes in the topology.

To me, this also means avoid running a remote provider add-in on a SharePoint web front end server, even though it is in a separate IIS web site. A remote provider add-in web application should run on a separate Windows server. Depending on the expected workload this could be a single server or even a high availability setup with multiple servers.

By the way, the inspiration for writing this article was a project involving a sub site creation add-in. During that project I learned a lot about the typical issues and solutions when routing the solution through the DTAP environment after our developer released it.

The add-in overrides the standard behavior of creating sub sites, by redirecting users to a remote form page on the remote web server. From there they can choose from a number of xml-based site templates (in this case based on the "PnP-Provisioning-Schema") which are uploaded and stored in a library on the root web within the SharePoint site collection where the add-in was installed. So depending on the number of uploaded site templates, users can choose what type of sites to create.

Although this particular case was about a specific type of add-in, the guidance in this article relates to any type of high trust provider hosted add-in deployed on-premises.

Environment setup

Let's start with an overview of the environment through a simplified diagram as shown in Figure 1:

a. There's a SharePoint Server (2013 or later) topology which has the app management service properly setup (plenty of guidance on this on the web);
b. A separate Windows Server is deployed to host one IIS website including one or more web applications;
c. In this case there is one IIS website which hosts the web application part of the provider hosted add-in in a virtual directory directly under the root of the website (just in case you need to host multiple web applications);
d. SharePoint and the add-in are connected through a trust which is explained later in this article;
e. Obviously the "app" package for the add-in is uploaded to the app catalog to support distribution and versioning;
f. From the app catalog, the add-in is installed in the site collection.

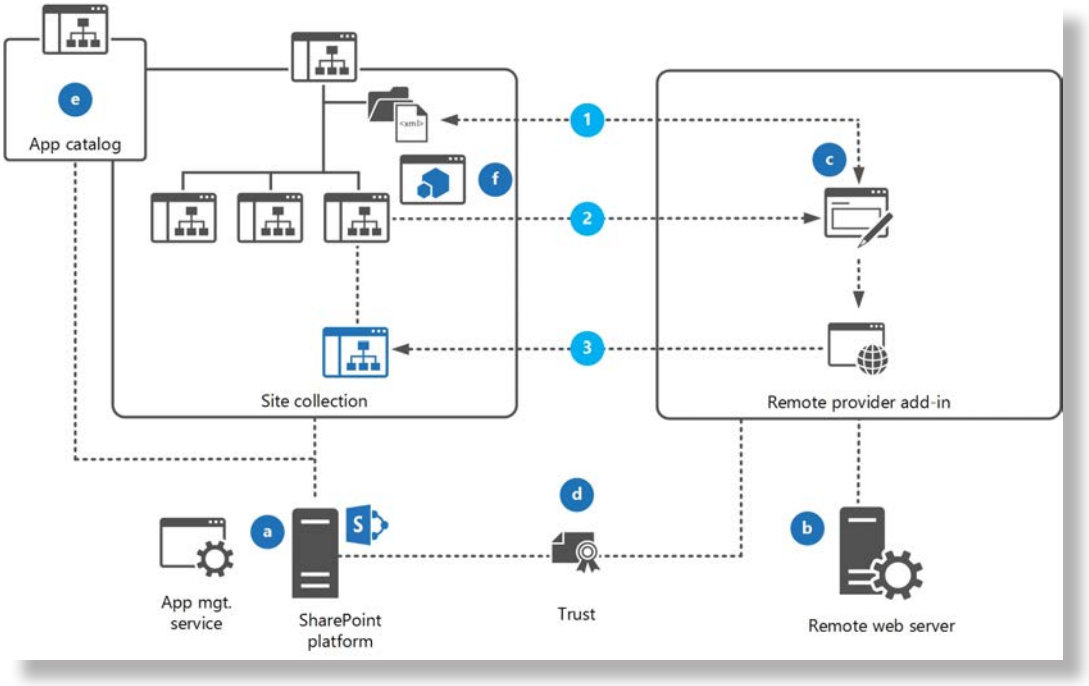


Figure 1: environment setup

During the installation event from the app catalog, the add-in kicks in:

1. It programmatically creates a standard document library to store the xml-based site templates;
2. The default "new subsite" link ("/_layouts/15/newsbweb.aspx") is overwritten upon page load using JSLink, redirecting users to the remote application;
3. When users provide the site title, Url, description and template choice, the sub site is then programmatically created in SharePoint from the remote application.

Our shopping list

So what was needed to set this up:

- A separate Windows server (2008 R2 or more recent) with decent (virtual) hardware specifications depending on the expected workload;
- It needs to be configured with the web- and application server roles and running .Net Framework 4.5;

- The Web Deploy software package may be installed to simplify the deployment of the remote application part;
- A server certificate to bind to the IIS website for secure communications between the remote web application and the end user;
- A client signing certificate to setup the trust between SharePoint and the remote web application.

Furthermore, the IIS website hosting the remote web application is configured to run on secure port 443 using the server certificate and having the built-in application pool identity as the identity for the application pool.

Production-ready High Trust

The remote application communicates with SharePoint using a client ID and issuer ID.

Client IDs are unique for each add-in and set in the add-in's manifest.xml and the web configuration in the remote web application part.

The issuer ID is generated when setting up a new trusted security token issuer using the client signing certificate. There is proper guidance on this on MSDN ("Create high-trust SharePoint Add-ins").

If configured correctly, the farm will have a brand new trusted security token issuer with a unique identity. Use PowerShell to request the Issuer ID.

```
Get-SPTrustedSecurityTokenIssuer | Select RegisteredIssuerName
```

Listing 1: Request trusted security token issuer

The result of running the cmdlet shows a long string, divided in two parts by the at (@) sign.

```
a827de45-7f34-4ecd-994f-c9e2de615d60@a1e21253-d089-45a7-a8c6-8334377ca352
```

Listing 2: Result of trusted security token issuer

The first part of the string contains the issuer ID and the second part (after the @ sign) the farm ID. So far, so good. Nothing new here that has not yet been explained before.

Now it becomes more interesting. There are a lot of articles on the web explaining on how to implement a high trust setup, but most cover the approach of exporting the trust certificate to a Personal Information Exchange file (.pfx) which includes the private key. That file is stored on the file system of the remote web server and the web configuration file contains a reference to this file and the password to access it.

```
<appSettings>
    <add key="ClientId" value="ad2498ee-867d-41f3-ae8f-43b8523af0f4" />
    <add key="ClientSigningCertificatePath" value="D:\Certs\MyCert.pfx" />
    <add key="ClientSigningCertificatePassword" value="@TrU89$t@" />
    <add key="IssuerId" value=" a827de45-7f34-4ecd-994f-c9e2de615d60" />
</appSettings>
```

Listing 3: High trust setup using a Personal Information Exchange file

From a security perspective, this is probably not the best setup for a production scenario as the Client signing certificate password is shown in the web configuration file.

A (better) alternative would be to store the trust certificate into the local Windows certificate store on the remote web server and reference it from the web configuration file.

```
<appSettings>
    <add key="ClientId" value="ad2498ee-867d-41f3-ae8f-43b8523af0f4" />
    <add key="ClientSigningCertificateSerialNumber"
value="281A030622ADC1B44DD42C2C37B004D4" />
    <add key="IssuerId" value="a827de45-7f34-4ecd-994f-c9e2de615d60" />
</appSettings>
```

Listing 4: High trust setup using a certificate store reference

This aligns with some common practices of distributing certificates directly to the certificate store of a server and avoids having .pfx file and password references in the web configuration file of the remote web application. The certificate should be placed in the "personal" certificate store.

To support this in the web application, a small change needs to be made to the TokenHelper file. Again, this is properly explained on MSDN ("Package and publish high-trust SharePoint Add-ins"). The TokenHelper is changed to reference the client signing certificate in the local Windows certificate store using its unique serial number.

To get this serial number, it seems obvious to copy it directly from the certificate details panel in the Windows certificate store.

There's the first pitfall: MSDN warns about this approach as spaces and hidden characters may be included in the clipboard resulting in an incorrect value for the "ClientSigningCertificateSerialNumber" node in the web configuration file.

When encountering this issue, I copied the serial number to a text editor. Visually it showed 32 characters, while the character count showed 33. Go figure!

A better approach is to use Windows PowerShell on the remote web server to retrieve it.

```
Get-ChildItem Cert:\LocalMachine\My | Select SerialNumber, Subject
```

Listing 5: Get the client signing certificate serial number through PowerShell

The output of this cmdlet shows a listing of certificate serial numbers and subjects making it easy to choose from.

```
SerialNumber                              Subject
------------                              -------
281A030622ADC1B44DD42C2C37B004D4          CN=apps.contoso.net
17B020B17C399DAF45CC96D955F631D8          CN=localhost
3B2A237D49C5EEBB4746305303EF9C7B          CN=WorkflowOutbound
608C1EA2443BCA88435B60AF7450C167          CN=AppServerGeneratedSBCA
0B80B00431F28794409D275E3FF950FA          CN=ForefrontIdentityManager
```

Listing 6: Client signing certificate serial number

In this example we needed the serial number for the certificate with the subject "CN=apps.contoso.net". No spaces or hidden characters here, so we're good to go!

So that's one pitfall out of the way. But there's more. When SharePoint communicates with the remote web application, the application pool identity plays a part in the authentication process.

The IIS application pool identity does not have default access to the local Windows Certificate store. This is required as this identity needs to be able to retrieve the private key from the client signing certificate for authentication purposes.

In IIS, the application pool should be configured with the setting "loadUserProfile = True" as the application pool may need to store temporary data and requires a cryptographic context when accessing the local Windows Certificate store.

Last but not least, access for the application pool identity to the private keys of the certificate need to be explicitly set. Windows PowerShell to the rescue.

```
[CmdletBinding()]
#Parameters for this script
Param(
    [Parameter(Mandatory=$True,Position=1)]
    [string]$serialNumber,
    [Parameter(Mandatory=$True,Position=2)]
    [string]$account,
    [Parameter(Mandatory=$True,Position=3)]
    [string]$rights
)

#Import PSSnapIn and assemblies
Import-Module webadministration

#Check if certificate exists
Try
{
    $WorkingCert = Get-ChildItem cert:\LocalMachine\My | Where-Object {$ _ .
SerialNumber -eq $serialNumber} | Sort-Object $ _ .NotAfter -Descending |
Select-Object -first 1 -erroraction STOP
    $TPrint = $WorkingCert.Thumbprint
    $rsaFile = $WorkingCert.PrivateKey.CspKeyContainerInfo.
UniqueKeyContainerName
}
Catch
{
    Write-Host “Unable to locate certificate for $($certSerialNumber)”
-ForegroundColor Yellow
    Exit
}

#Find permissions for the object
$keyPath = ‘C:\ProgramData\Microsoft\Crypto\RSA\MachineKeys\’
$fullPath = $keyPath+$rsaFile
$acl = Get-Acl -Path $fullPath
$permission = $account,$rights,’Allow’
$accessRule = New-Object System.Security.AccessControl.FileSystemAccessRule
$permission
$acl.AddAccessRule($accessRule)

#Attempt to set permissions for the object
Try
{
    Set-Acl $fullPath $acl
    Write-Host $rights’ permissions set for ‘$account’ on certificate with’
$WorkingCert.Subject -ForegroundColor Green
}
Catch
{
    Write-Host ‘Unable to set’$rights’ permissions for ‘$account’ on
certificate with’ $WorkingCert.Subject -ForegroundColor Yellow
    Exit
}
```

Listing 7: Set permissions for application pool account

The “set-CertPerms” script takes the following parameters:

- SerialNumber: the serial number of the client signing certificate used in the trust setup;
- Account: the identity of the IIS application pool of the remote web application;
- Rights: the type of access; “read” is sufficient for the intended setup.

```
.\set-CertPerms.ps1 -SerialNumber 281A030622ADC1B44DD42C2C37B004D4 –account
“IIS AppPool\apphost” –rights “read”
```

Listing 8: Set permissions for application pool account

If executed correctly, the application pool identity (“IIS AppPool\apphost” in the example) should have read permissions to the private keys of the client signing certificate stored in the local Windows Certificate Store of the remote web server. You can validate the permissions by opening a certificate management console, browsing to “Personal > Certificate”, right-clicking the certificate and choosing “All Tasks > Manage Private Keys”.

Ready to deploy

So all is ready then roll out the web application through a Web deploy package on the remote server and to upload the "app" package to the app catalog. The nice thing about a web deploy package is that it comes with a "…SetParameters.xml" file in which you can set the deployment settings and locations for a particular environment. MSDN has some guidance on this ("Howto: Create a Web Deployment Package in Visual Studio"). It's worth to explore this approach in more depth.

By deploying the web deploy package, the web application in the IIS website should be configured as expected including authentication settings. If deployed correctly, the web application should be configured with "Windows Authentication" enabled ("NTLM" should be listed above "Negotiate" in the "Providers" section) and "Anonymous" disabled. If the manifest file references a remote web services in the "InstalledEventEndpoint", the virtual directory containing that web service needs to be set to anonymous only (disabling "Windows Authentication").

As the high trust setup requires many small steps, it's easy to make mistakes (I have lots of experience in that department). But in the end most issues occurring during the setup can be related to:

- Meeting all prerequisites;
- The configuration of the trust;
- Using the right client en issuer id's;
- Having the proper start page and endpoint Url's in the add-in's manifest.xml (this can be cumbersome in an DTAP setup);
- Application pool, authentication and (certificate) binding settings in IIS;
- and the high trust setup described in this article.

Conclusion

This article was meant to outline some interesting pitfalls for implementing high trust add-ins in a full on-premises scenario. Please feel free to contact me with feedback or suggestions.

Developing Office 365 apps with the Microsoft Graph and Office UI Fabric

by Maarten Stelling

Last November, Microsoft announced the global availability of the Microsoft Graph, formerly known as Office 365 Unified API. A single API endpoint for accessing data out of the Microsoft cloud including Office 365 and Azure. This single endpoint gives developers the ability to access data from multiple services within Office 365.

This becomes very useful when it comes to integrating data into an Office 365 app or SharePoint add-in. Combined with the Office UI Fabric framework, this new set of tools gives developers the power to quickly create their custom applications that fit seamlessly into the Office 365 platform. We'll build a small "Who is Who" Office 365 app which uses the Microsoft Graph and the Office UI Fabric framework for real world scenarios.

Creating the Office 365 app

In this example we'll be creating a plain MVC application, but since the Microsoft Graph uses open standards like OAuth 2.0, REST and JSON every development framework can be used. This includes PHP, Java, Ruby and single page application frameworks such as AngularJS. I'm not going to explain every step for creating this app, just the most important ones. The whole source code is available for download on GitHub.

When creating the application we'll set the authentication mode to No Authentication, since we'll be setting up the authentication ourselves. The application will be hosted in Azure so I'll turn on the setting Host in the cloud.

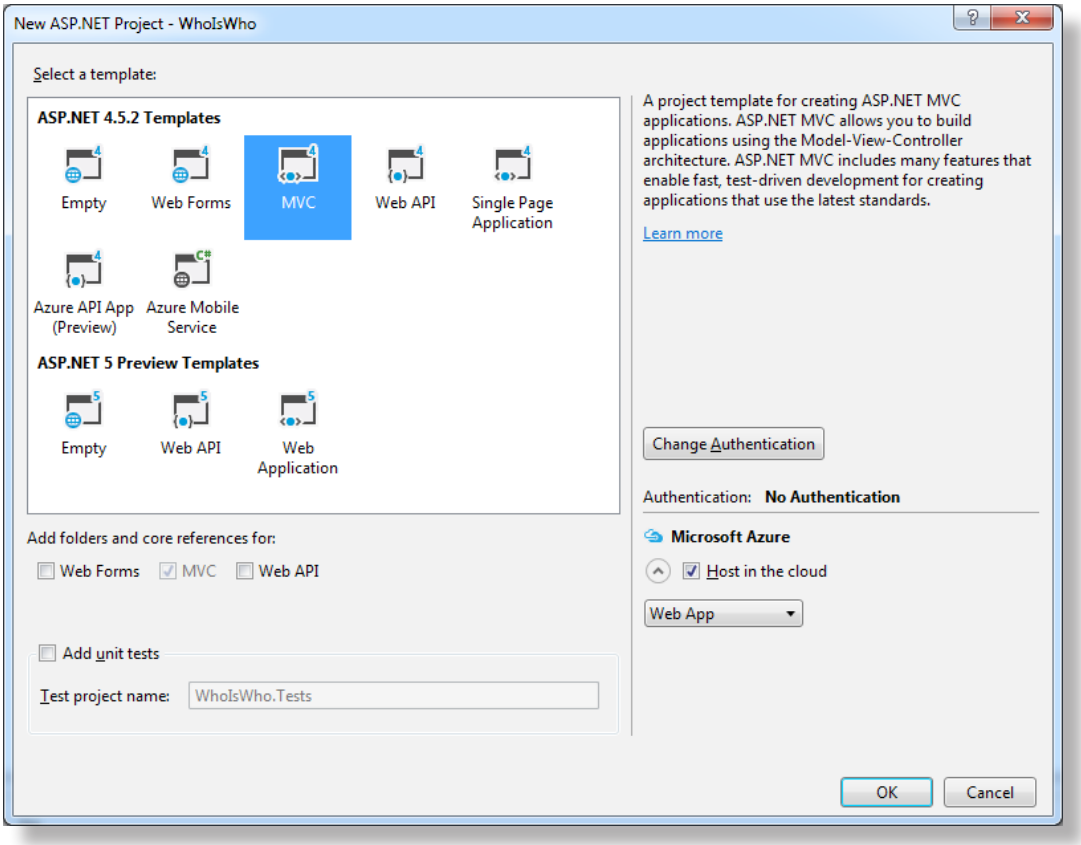


Figure 1: Creating the Office 365 app

Using Azure AD for authentication

Since the Microsoft Graph uses Azure Active Directory for authentication, we must first register the application in Azure AD. Go to the Microsoft Azure portal and open the Azure AD used for the Office 365 tenant where we will be using our app. Now click the Application tab and use the add button at the bottom of the screen.

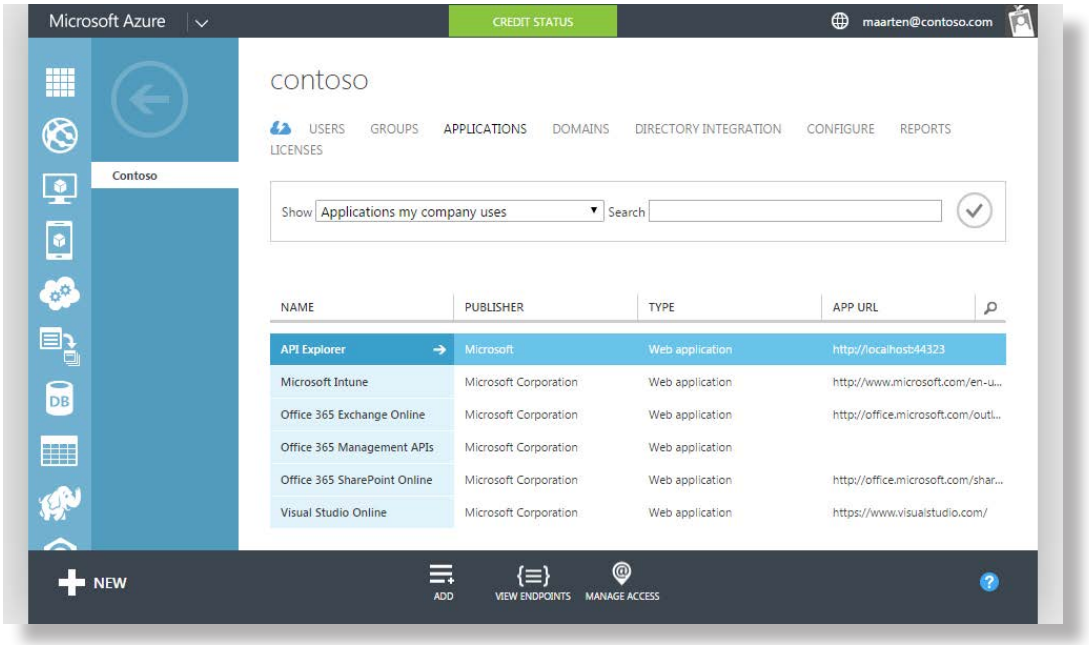


Figure 2: Add the new application to Azure AD

In the popup window choose Add an application my organization is developing.

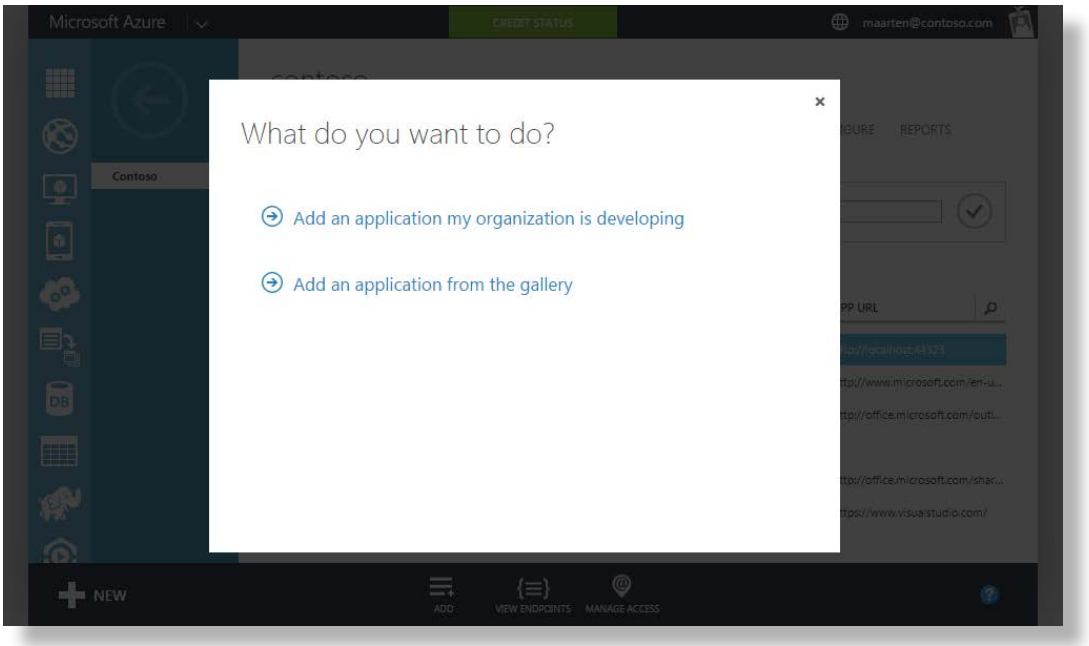


Figure 3: Add an application my organization is developing

Name the application and choose Web application and/or web API as the Type.

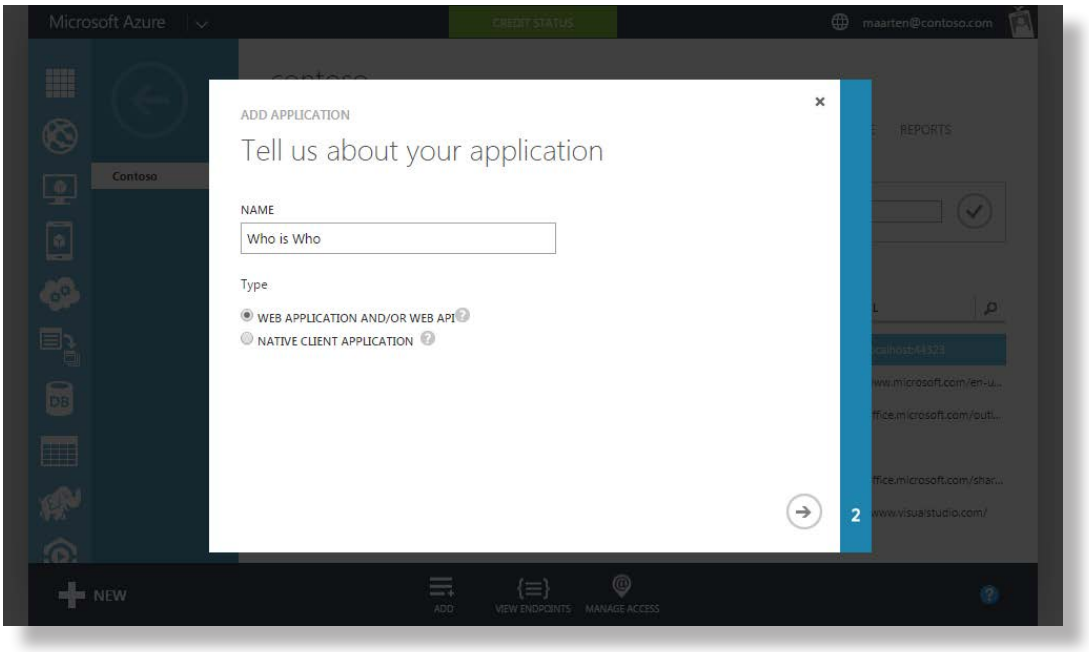


Figure 4: Name the application

Fill out the Sign-on URL of the application, usually the ordinary URL of the web application, and fill out the unique App ID URI.

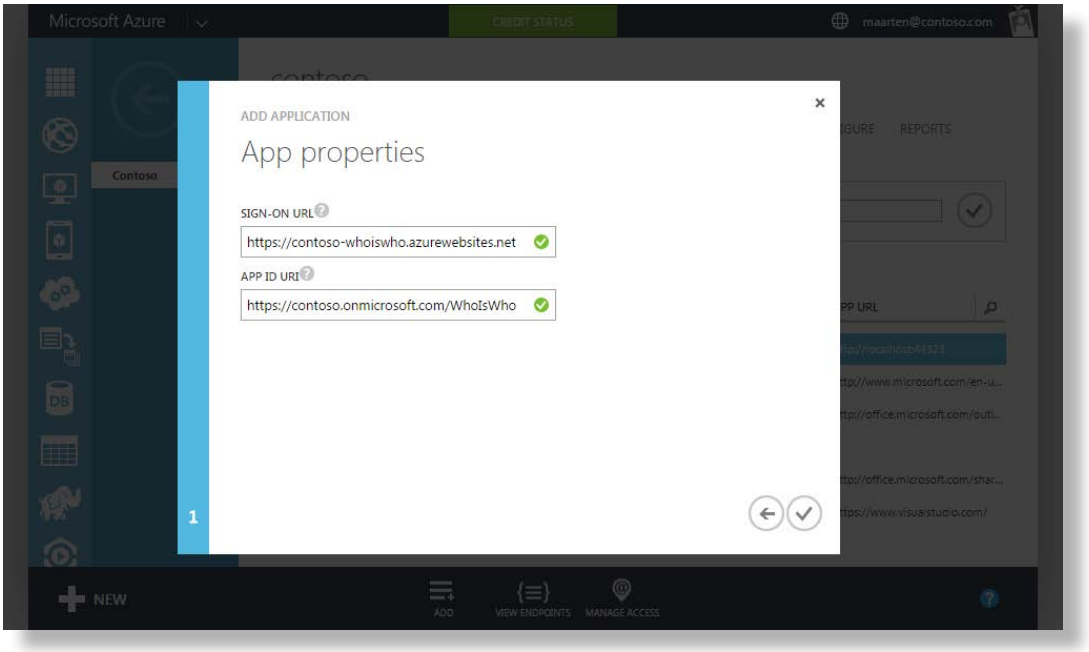


Figure 5: Add URI's

When the application is created, open the configure tab and scroll down to the section where the Client ID is displayed. Copy the Client ID and note it down. Go create a new key, also referred to as Client Secret, in the Keys section and save the application configuration. After saving, the newly created key will be visible. Make sure you copy and note down this key as well, as you won't be able to retrieve it after you leave this page.

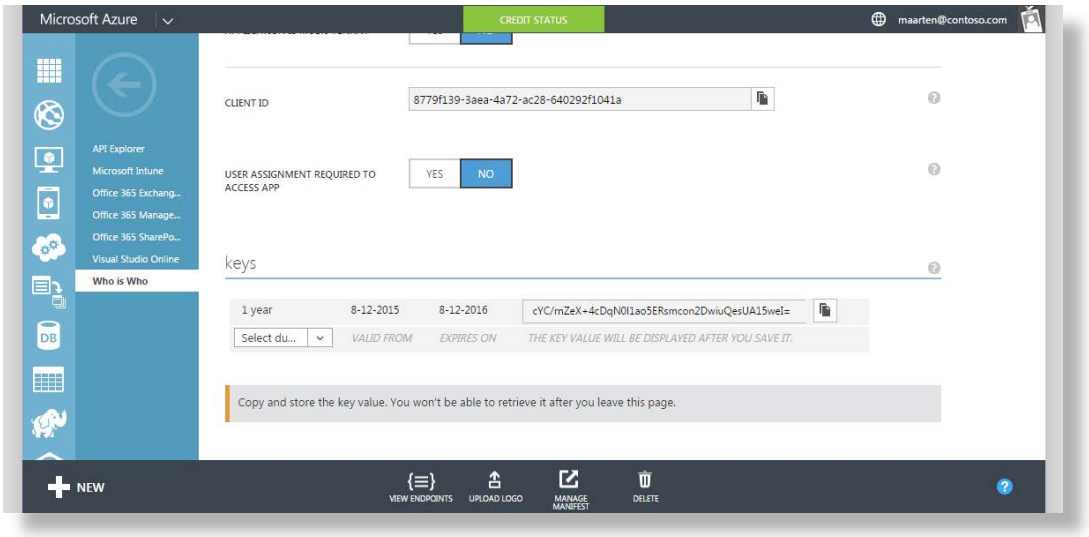


Figure 6: Create and save application keys

A best practice would be saving the keys directly into the web.config of your application as shown in listing 1.

```
<appSettings>
  <add key="ClientID" value="YOUR_CLIENT_ID" />
  <add key="ClientSecret" value="YOUR_CLIENT_SECRET" />
</appSettings>
```

Listing 1: Application configuration

Since we want to give our app permission to use the Microsoft Graph, we must set this up in the Permissions to other applications section. Scroll down to the bottom of the page, click the Add application button and select the Microsoft Graph in the popup window.

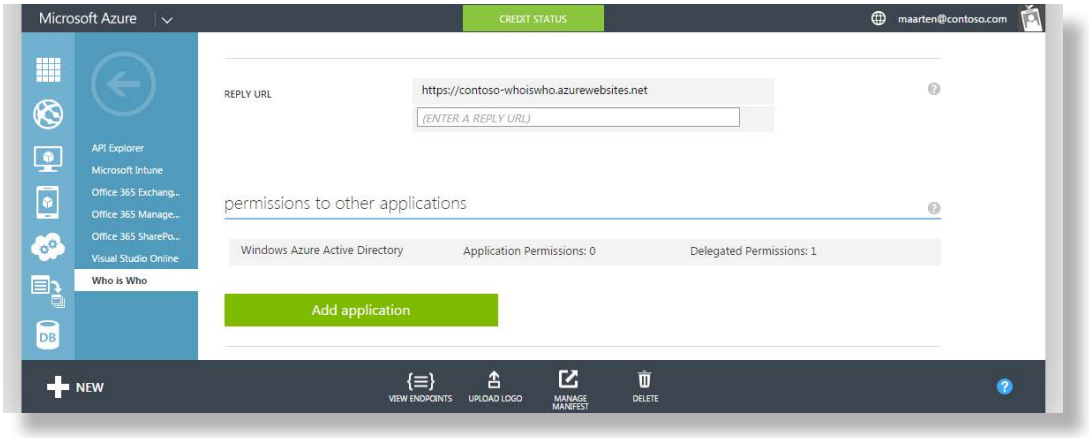


Figure 7: Add permissions to application

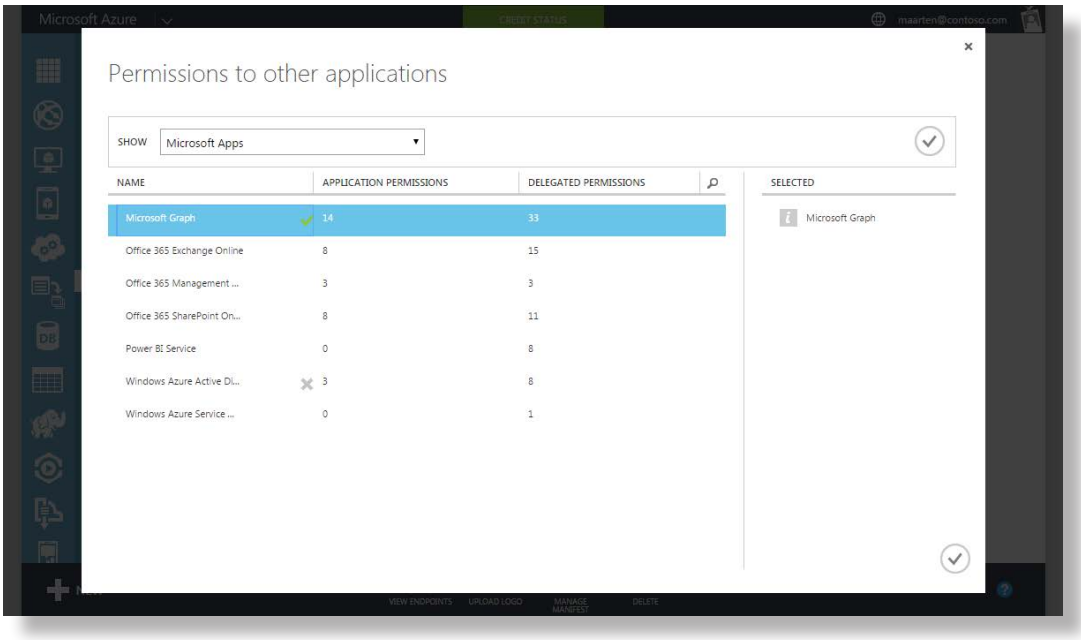


Figure 8: Add permissions to the Microsoft Graph

After adding the Microsoft Graph application we must select the appropriate permissions our app is going to need in order to be able to access the correct data. Click open the Delegated Permissions select box and select the "Read all users' full profiles" permission level, then save the application again.

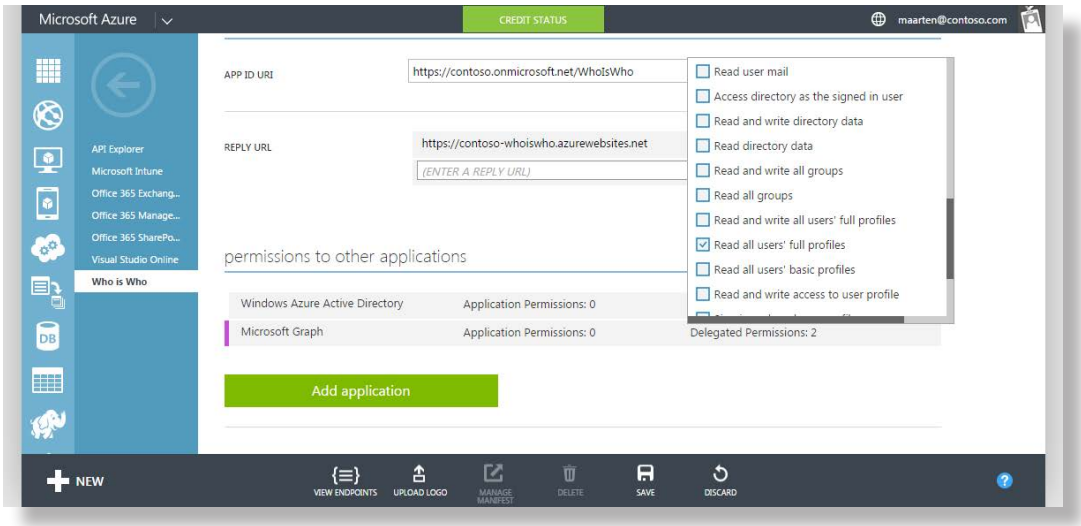


Figure 9: Select permission levels

Last step is giving our user account access to the app by adding the user assignment. Go to the Users tab, select the appropriate user account and click the Assign button at the bottom of the screen.

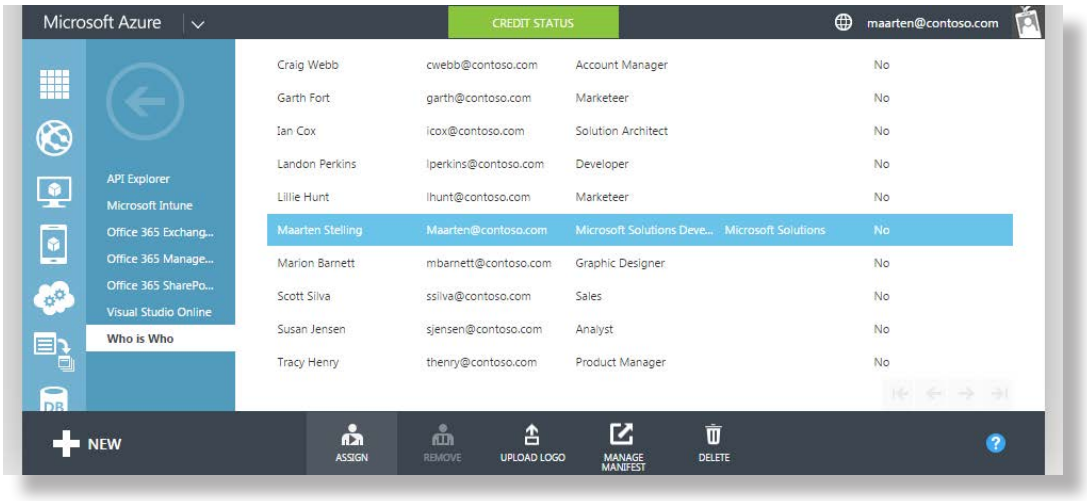


Figure 10: Assign user account access

When the appropriate users are given access, the application is successfully registered in Azure AD.

Obtaining data from the Microsoft Graph

Now that we have registered our application in Azure AD, it's time to get some data out of Office 365 using the Microsoft Graph API. First, we must add the Active Directory Authentication Library (ADAL) NuGet package to our MVC application in order to set up the authentication. Open up the Package Manager Console:

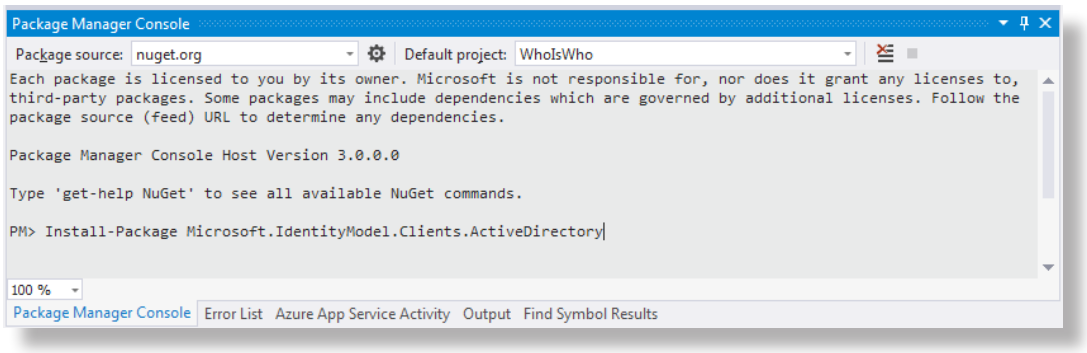


Figure 11: Install Active Directory Authentication Library

We'll be using some helper classes in our application. As a reference, figure 12 shows the folders and classes we'll be using.

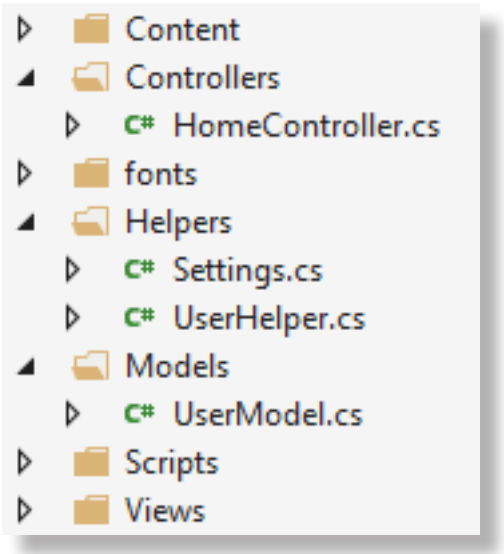


Figure 12: Application classes

Authenticate the user

We want to authenticate the current user so we'll be able to call the Microsoft Graph. In our main action we're checking if the current user already has an access token. If not, the user gets redirected to the SignIn action. For this we'll need to create the SignIn action in our Home controller. This action will be used to redirect the app to the Azure AD authorization request URL and retrieve an authorization code.

```
public ActionResult Index()
{
  // Check if current user has an access token
  // If not, redirect to sign in action
  if (String.IsNullOrWhiteSpace((string)Session["access _ token"]))
  {
    return RedirectToAction("SignIn");
  }

  // Get users out of Office 365 using the Microsoft Graph
  List<UserModel> users =
  UserHelper.GetUsers((string)Session["access _ token"]);

  return View(users);
}

public ActionResult SignIn()
{
  // Create new authentication context
  var authContext = new AuthenticationContext(Settings.Authority);

  // The URL where Azure redirects to after successful login
  Uri loginRedirectUri = new Uri(Url.Action(
    "Authorize",
    "Home",
    null,
    Request.Url.Scheme));

  // Parameterized URL for Azure login
  Uri authUri = authContext.GetAuthorizationRequestURL(
    Settings.GraphUri,
    Settings.ClientId,
    loginRedirectUri,
    UserIdentifier.AnyUser,
    null);

  // Redirect to login page, then come back to the Authorize method
  return Redirect(authUri.ToString());
}
```

Listing 2: Set up the main and sign in action

```
public class Settings
{
  public static string ClientId = ConfigurationManager.
AppSettings["ClientID"];
  public static string ClientSecret = ConfigurationManager.
AppSettings["ClientSecret"];

  public static string Authority = @"https://login.microsoftonline.com/
common";
  public static string GraphUri = @"https://graph.microsoft.com/";
}
```

Listing 3: Settings helper class

After the user successfully signs in, we must create an Authorize action in our Home controller to request the access token from Azure AD. We'll save the returned access token into the session so we can use it for multiple requests.

```
public async Task<ActionResult> Authorize()
{
  // Create new authentication context
  var authContext = new AuthenticationContext(Settings.Authority);

  // The URL where Azure redirects to after successful login
  Uri loginRedirectUri = new Uri(Url.Action(
    "Authorize",
    "Home",
    null,
    Request.Url.Scheme));

  // Get the access token
  var authResult = await authContext.AcquireTokenByAuthorizationCodeAsync(
    Request.Params["code"],
    loginRedirectUri,
    new ClientCredential(Settings.ClientId, Settings.ClientSecret),
    Settings.GraphUri);

  // Save the token in the session.
  Session["access _ token"] = authResult.AccessToken;

  return RedirectToAction("Index");
}
```

Listing 4: Get the access token using the authorization code

Using the access token in a request to the Microsoft Graph API

In our "Who is Who" application we want to request a list of users within the Office 365 tenant and show them in a grid page. For this we need to make an authenticated call with the access token to the Microsoft Graph API. In listing 2 we're calling the GetUsers method, listing 5 displays how this method is implemented in the UserHelper class.

```
public class UserHelper
{
  public static List<UserModel> GetUsers(string accessToken)
  {
    List<UserModel> users = new List<UserModel>();
    MediaTypeWithQualityHeaderValue Json = new
    MediaTypeWithQualityHeaderValue("application/json");

    using (var client = new HttpClient())
    {
      using (var request = new HttpRequestMessage(HttpMethod.Get,
      @"https://graph.microsoft.com/v1.0/users"))
      {
        request.Headers.Accept.Add(Json);
        request.Headers.Authorization = new
        AuthenticationHeaderValue("Bearer", accessToken);

        using (var response = client.SendAsync(request).Result)
        {
          if (response.StatusCode == HttpStatusCode.OK)
          {
            var json =
            JObject.Parse(response.Content.ReadAsStringAsync().Result);

            foreach (JToken user in json.SelectToken("value").Children())
            {
              users.Add(new UserModel(user));
            }
          }
        }
      }
    }
    return users;
  }
}
```

Listing 5: Calling the Microsoft Graph API endpoint

The result string gets parsed into a JSON object and every returned user is then used to create a custom user model as shown in listing 6. The complete list of users gets returned to the action and eventually to the view.

```
public class UserModel
{
  public UserModel(JToken user)
  {
    this.Name = (string)user[“displayName”];
    this.JobTitle = (string)user[“jobTitle”];
    this.Email = (string)user[“userPrincipalName”];
    this.Phone = (string)user[“mobilePhone”];
    this.OfficeLocation = (string)user[“officeLocation”];
  }

  public string Name { get; set; }
  public string JobTitle { get; set; }
  public string Email { get; set; }
  public string Phone { get; set; }
  public string OfficeLocation { get; set; }
  public byte[] Photo { get; set; }
}
```

Listing 6: User model class

And that’s how we call the Microsoft Graph API endpoint from our application. You can see it only takes a couple of lines of code to get data out of Office 365 and return it to our view. Now that we have a list of users available, let’s see how we can give it a nice look and feel.

Applying the Office look and feel with Office UI Fabric

Many developers are struggling with giving their custom applications a nice interface, especially when it comes to developing a SharePoint add-in or Office 365 app. The custom application just doesn’t seem to fit in with the whole platform.

Last August, Microsoft released a powerful framework called Office UI Fabric. Office UI Fabric is a responsive, mobile-first, front-end framework, designed to make it simple to create web experiences using the Office Design Language. With Office UI Fabric you’re able to apply styling and use components to make your applications look and feel like the rest of Office.

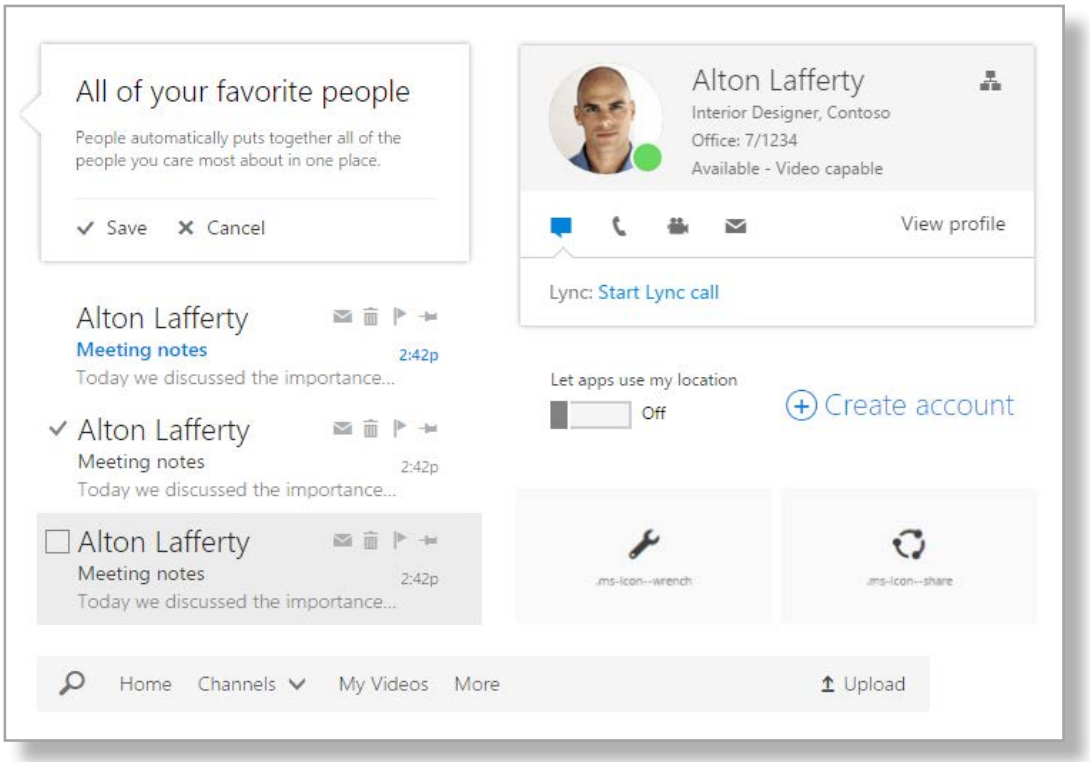


Figure 13: Office UI Fabric example styling and components

The styling takes into account typography, color, icons, animations, responsive grid layouts and localization. There are reusable components such as input, layout, navigation and content (persona card, list item and table views). Figure 13 displays some styling and components examples. The framework is used internally on products within Office 365 - such as the suite branding, OneDrive.com, Outlook.com, Delve and the Video Portal.

For our "Who is Who" application we'll be using the Persona, PersonaCard, NavBar and SearchBox components along with some typography and icon styling. Since we've registered our application in the Office 365 tenants Azure AD the application is available within Office 365 as an app. I've added the app to my app launcher which looks like figure 14.

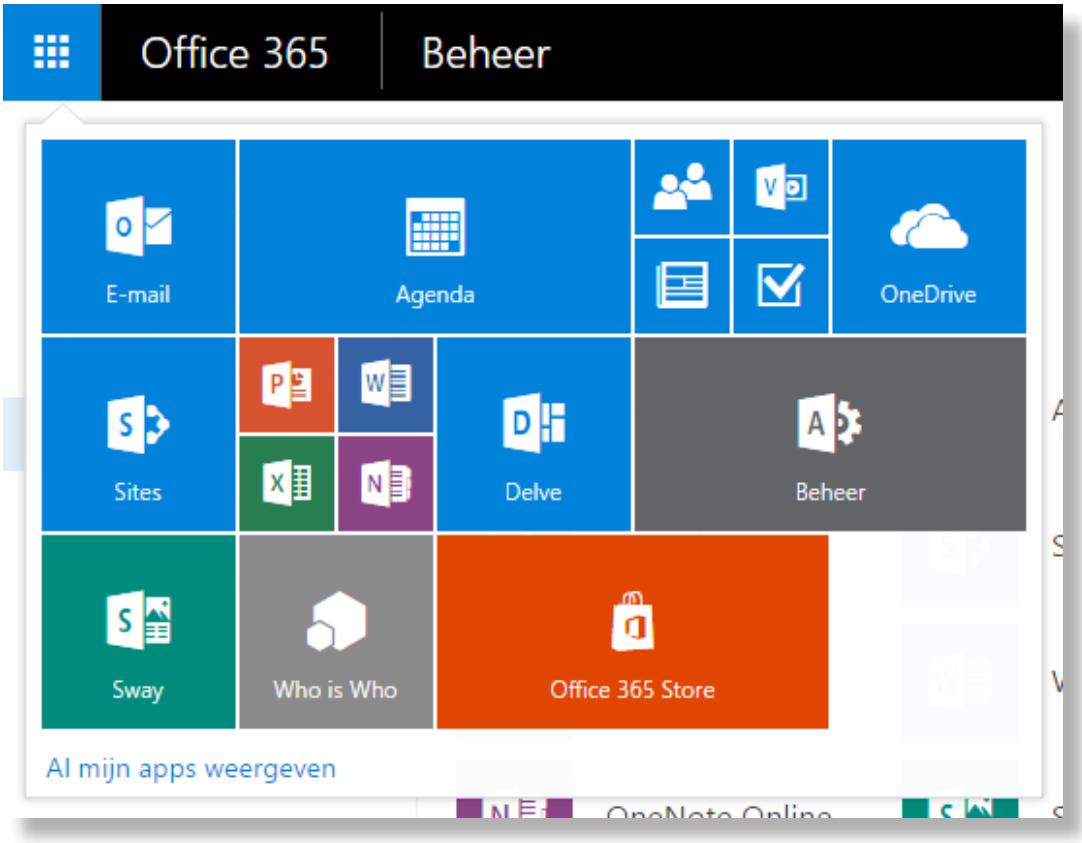


Figure 14: Office 365 app launcher

Clicking the "Who is Who" tile opens up the application which looks like figure 15.

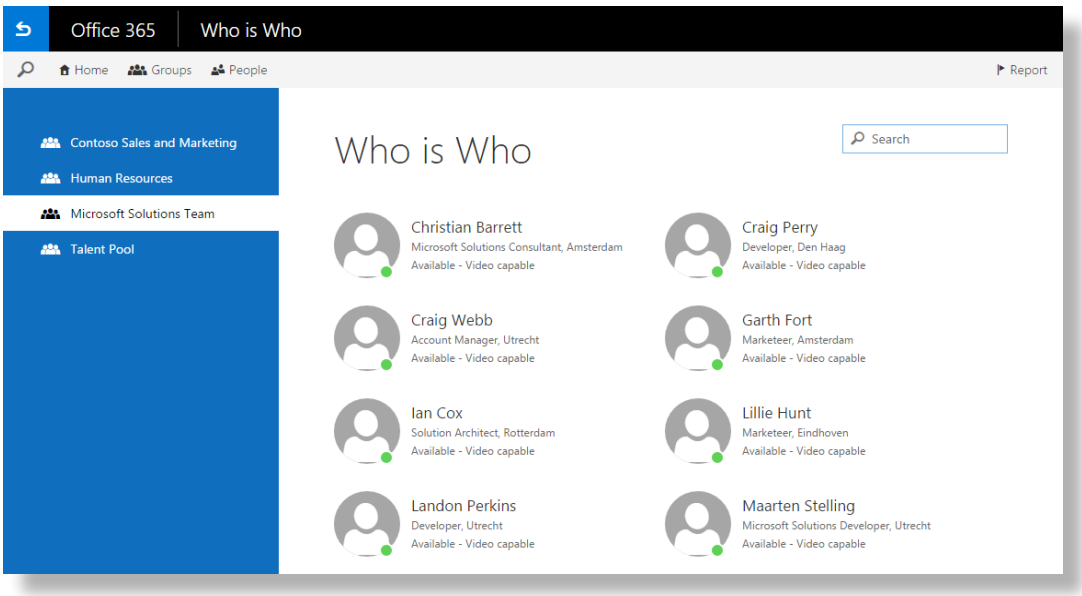


Figure 15: Who is Who application screenshot

Using the Office UI Fabric framework

Integrating the Office UI Fabric framework into a web application is very straightforward. For a detailed description visit the documentation website http://dev.office.com/fabric/getting-started. First, we must download the latest release and add the necessary CSS and JavaScript files to our application. Next, we must add the references to the stylesheets in the head of our HTML document.

```
<link rel="stylesheet" href="~/Content/fabric.min.css" />
<link rel="stylesheet" href="~/Content/fabric.components.min.css" />
```

Listing 7: Referencing the Office UI Fabric stylesheets

Since the components we'll be using make use of additional JavaScript we'll reference those files as well. Make sure jQuery is loaded as well.

```
<script src="~/Scripts/Jquery.PersonaCard.js" type="text/javascript"></script>
<script src="~/Scripts/Jquery.SearchBox.js" type="text/javascript"></script>
```

Listing 8: Referencing the Office UI Fabric javascript

Now that we have our CSS and JavaScript files in place we're ready to create the HTML. For setting up the header and search box component we're simply adding the HTML provided by Office UI Fabric into our page, which is the Index view from our Home controller.

```
<h1 class="ms-font-su">Who is Who</h1>
<div class="ms-SearchBox">
  <input class="ms-SearchBox-field" type="text">
  <label class="ms-SearchBox-label">
    <i class="ms-SearchBox-icon ms-Icon ms-Icon--search"></i>Search
  </label>
  <button class="ms-SearchBox-closeButton">
    <i class="ms-Icon ms-Icon--x"></i>
  </button>
</div>

<script type="text/javascript">
$(function () {
  $('.ms-SearchBox').SearchBox();
});
</script>
```

Listing 9: Heading and search box HTML components

When adding a predefined class to the H1 heading, this element will automatically get the correct typography from the Office UI Fabric stylesheets. The search box HTML is fully provided by the Office UI Fabric framework and can be copy pasted into our application. In order to add a nice focus/blur effect, just call the SearchBox() method on the search box element and Office UI Fabric will take care of the rest.

User grid

We'll be using the Persona component to set up the user grid. Since our view contains a Model with a list of users, we'll just iterate through the Model and set up the Persona HTML for each user.

```
@model IList<WhoIsWho.Models.UserModel>

@foreach (var user in Model)
{
  <div class="ms-Persona ms-Persona--lg">
    <div class="ms-Persona-imageArea">
      <i class="ms-Persona-placeholder ms-Icon ms-Icon--person"></i>
    </div>
    <div class="ms-Persona-presence"></div>
    <div class="ms-Persona-details">
      <div class="ms-Persona-primaryText">@user.Name</div>
      <div class="ms-Persona-secondaryText">@user.JobTitle, @user.
OfficeLocation</div>
      <div class="ms-Persona-tertiaryText">Available - Video capable</div>
    </div>
  </div>
}
```

Listing 10: Office UI Fabric Persona components

We want to create a fancy effect for when someone hovers over one of the cards, so we add the HTML markup of the PersonaCard for each user as well. With a few lines of custom JavaScript we set up the hover event so the PersonaCard pops up and the outcome will look like figure 16.

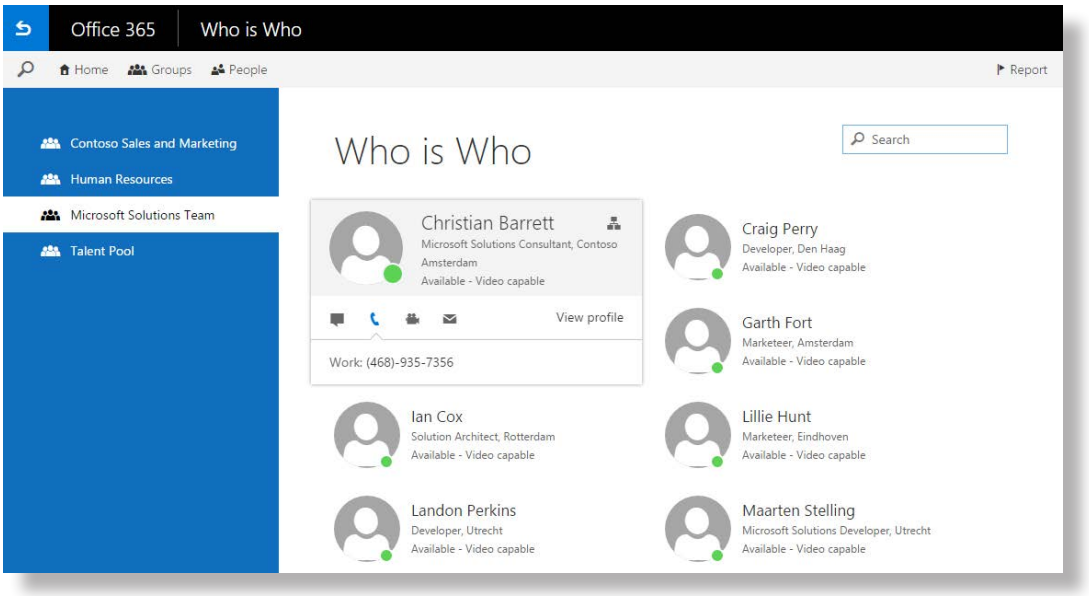


Figure 16: PersonaCard hover effect

And that's how we create an Office 365 app which uses the Microsoft Graph and Office UI Fabric capabilities. You must take into account that I've added some extra CSS and JavaScript to make the application look like figure 16. I can imagine that end-users expect to see the profile pictures of all of the users as well. The Microsoft Graph API offers an endpoint for getting the photo per user. This method could be called from the user model constructor or fully client side with JavaScript. You could try integrating the profile picture as an example, happy coding!

Conclusion

As we can see within our sample application, with this new set of tools Microsoft really offers developers the ability to create powerful applications that can be used within the Microsoft platforms. With the Microsoft Graph we can retrieve data from, and integrate into, our custom application which comes from different Microsoft cloud services with just a couple lines of code. By using the Office UI Fabric framework the application gets the Office look and feel and makes it fit perfectly in the Microsoft platform.

Links

- More information about the covered topics:
- Microsoft Graph API (https://graph.microsoft.com).
- Office UI Fabric (http://dev.office.com/fabric).
- Azure AD (https://azure.microsoft.com/nl-nl/documentation/articles/active-directory-whatis).
- Who is Who app source code (https://github.com/maartenstelling/Diwug).

Welcome back old friend

by Jasper Oosterveld

For many years, SharePoint was always the center of attention. Ever since its release in 2007, the growth and popularity was spectacular. SharePoint had its own yearly event, free community driven Saturday events, local user groups and more conferences popping up every year, worldwide. Although these facts speak for themselves, there was something wrong. The brand name seemed to get less attention and love from Microsoft per year. I blame Office 365. Don't get me wrong, I love Office 365, but SharePoint became only a small part of a bigger picture. Just take a look at the Sites tile in the Office 365 App Launcher:

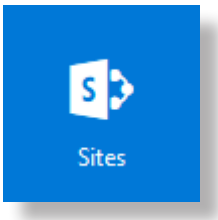


Figure 1: The Sites tile in the Office 365 App Launcher

We all recognize the SharePoint logo. People stared to worry: "Is SharePoint dying?", "Is SharePoint going away?". Well, after the now legendary online event called the Future of SharePoint we all know the answer:

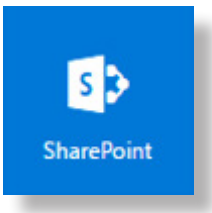


Figure 2: SharePoint is back!

Most definitely not! Welcome back old friend.

Vision & Innovation

Jeff Teper, the Godfather of SharePoint, laid down his and Microsoft's four areas of innovation for SharePoint and SharePoint on-premises:

- Simple and powerful file sharing and collaboration on any device.
- The mobile and intelligent intranet, with modern team sites, publishing and business applications on your desktop and in your pocket.
- An open and connected platform that evolves SharePoint extensibility to embrace modern web development.
- Investments in security, privacy and compliance across Office 365.

This quarter we can expect the following:

- Modern document library experience (currently rolling out to First Release tenants).
- SharePoint mobile app for iOS.
- SharePoint home in Office 365.
- Modern lists experience.
- Site activity and insights on the Site Contents page.

For the remainder of the year, we have to keep our eyes out for the following:

- SharePoint mobile app for Windows and Android.
- Integration of SharePoint sites and Office 365 Groups.
- Simple, fast site creation.
- Modern pages experience.
- Team and organizational news and announcements.
- PowerApps and Microsoft Flow integration with SharePoint.

We can all see that it's an exciting time for the SharePoint platform. It's great to see new services (PowerApps & Flow & Groups) coming together with SharePoint. My favorite development is the focus on basic features such as file creation and sharing, the launch of a native SharePoint App. And last but not least, the integration of team sites with Groups.

Back to basic

The announcements during the Future of SharePoint proved to me that Microsoft is going back to basics with SharePoint and really focussing on the strengths of the platform: collaboration between teams & individuals. In my eyes, there hasn't been much happening in the SharePoint space over the last couple of years. Microsoft had a huge focus on Office 365 and pushing it towards new and existing customers. Yes, SharePoint Online received new features, but nothing major. SharePoint 2013 contained nothing new compared to its online bigger brother. You have seen new services appearing, clearly inspired by SharePoint, such as Delve, Groups and the Video Portal. I totally understood this move because Microsoft wants to keep up with the competition and make Office 365 the platform for innovation and productivity. SharePoint always lagged behind. For example, transferring documents between sites, or Groups and OneDrive, has been a very painful process. Microsoft is now dedicated to really solving these types of issues and making the platform more powerful than before.

We are finally going to be able to easily move files between different services and sites:

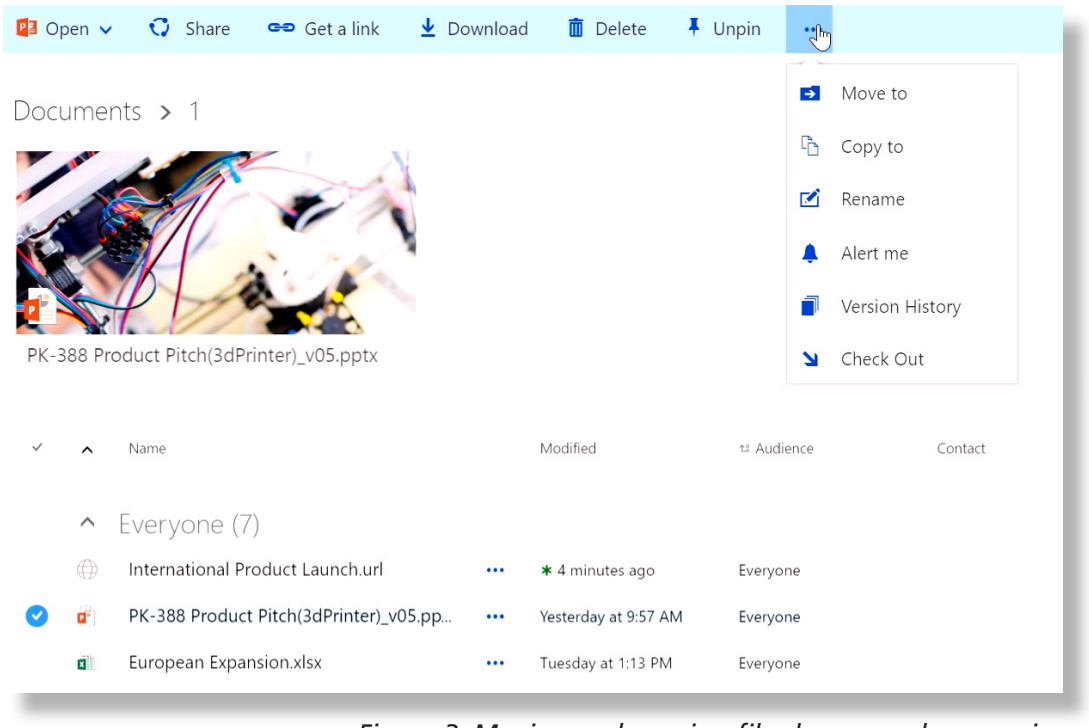


Figure 3: Moving and copying files has never been easier

You start working in OneDrive on your presentation for the upcoming team meeting. The presentation is aimed at a review of the Future of SharePoint. Wait a minute! Your colleague Dave also watched the event and you want his input. You

share the file with one click and work together on the presentation. Now it's time to share the presentation with your team. You have been working with Groups for a while now. Do you have to download and upload the document in the Group? No, with a couple clicks you easily move the file to the Group. You aren't using a Group? No problem, just move the file to your team site. Sounds like a basic feature right? Let's not forget this was an absolute pain in the behind for many, many years. Microsoft is dedicated to make your and your business users lives easier.

Embracing the new world

Finally, Microsoft released their very own SharePoint mobile App for all platforms, starting with iOS:

Figure 4: SharePoint App created by Microsoft

We are all part of the mobile device revolution. We have an iPhone (believe me, you should get one), Android or perhaps a Windows Phone (believe me, you shouldn't get one). I am almost certain you have a tablet as well. This isn't only the case for techies but for most business users it's a reality. We all want to use our mobile device to visit our corporate Intranet portal and work with our, and colleagues, content. What's the to-go platform? SharePoint of course. No more need for expensive 3rd party mobile Apps. The SharePoint App, created by Microsoft, is finally here. The App is going to be a game changer. The App is free, fast, integrated with the Office Graph and really focused on the new improvements for sites, libraries and lists. I have talked to the product team and they are working extremely hard to release a great App for SharePoint users all over the world. By using Jeff Tepers words: "Your Intranet in your pocket".

SharePoint on-premises

The majority of the projects I work on, are aimed at SharePoint Online. Over the last few years, we have seen a huge interest and increase in customers moving towards Office 365. That being said, there are still customers, for example law firms, who aren't ready yet for the Cloud revolution. They want to keep using SharePoint on-premises. This is mainly due to concerns about security from the business and IT departments wanting to keep control over their hardware & software. Of course, I don't entirely agree with this point of view but that's the reality and world we live in. These companies, of course, have every right to an innovating, business empowering and reliable SharePoint platform.

That's where SharePoint 2016 steps in. Microsoft built SharePoint Server 2016 from the Cloud up with all of the experiences and lessons learned from hosting SharePoint itself. The latest on-premises release is more powerful than before. Microsoft really shows its commitment by release feature packs, from 2017 onwards, to provide its customers with new features and innovations from SharePoint Online. This announcement is unbelievably exciting news and important. You can clearly

see that Microsoft is creating a bridge between SharePoint Online and on-premises and living up to the hybrid promise.

Team Sites & Groups: United we stand together

Funnily enough, my last article for the DIWUG e-magazine was about Groups. I discussed my five favorite features and hoped to inspire people looking into Groups. The evolution of Groups has been growing steadily since its release. One of the major pain points was the lack of integration with team sites. This caused a lot of confusion in the SharePoint & Office 365 community but mostly with our customers. What are you supposed to use and when? Microsoft did announce an integration at the Australian Ignite Conference last year but then it stayed quiet for a while. The missing piece, from a functional and business perspective, has always been connecting the two with each other. By realizing this connection, Microsoft will create a collaboration "tour de force" that is going to be unstoppable. The time is finally here! Are you creating a team site? You automatically receive a Group. Are you creating a Group? You automatically receive a Team Site. You are now able to leverage the power of team sites and Groups. The quick launch of a team site is extendible with quick links to, for example, conversations or Planner. Life has become a lot easier, believe me.

Conclusion

SharePoint is back, stronger than ever and here to stay. I haven't seen such commitment since the announcement of the "Cloud first" strategy during the 2012 SharePoint Conference. Is there any critique though? Yes, I am Dutch for a reason. My major concern is branding. The majority of our customers have a custom branding in their SharePoint Portals. How is this going to work with the new team site, document library and list experience and look & feel? The new responsive pages are incredible, but what about page lay-outs and pushing corporate fonts? I haven't heard any statements or information about these concerns. This is something that has to be addressed in the near future. I am definitely not going to close this article with critique but with overall praise. The roadmap, vision, innovation areas and overall proudness by Microsoft promises a very bright future for SharePoint. Online and on-premises.

Everything you need to know about cloud hybrid search

by Nico Martens

This article describes the new cloud hybrid search capabilities in SharePoint 2013 and SharePoint 2016. Use this article to configure cloud hybrid search in your organization and learn what you need to know.

If you need to decide whether to use or not use cloud hybrid search, please read my Cloud hybrid search considerations blog post before deploying cloud hybrid search: http://www.sharepointrelated.com/2016/03/02/cloud-hybrid-search-considerations/

What is cloud hybrid search

Before cloud hybrid search, Microsoft provided hybrid search scenarios between your on-premises SharePoint environment and SharePoint Online. These solutions were based on query federation. For instance, when you searched for a document in SharePoint Online in your on-premises environment, the query would be sent to the on-premises environment, and the results are returned back to the user in SharePoint Online. Microsoft released a script to automate this: http://blogs.msdn.com/b/spses/archive/2015/11/17/office-365-sharepoint-hybrid-configuration-wizard.aspx.

In September 2015, Microsoft released the new cloud hybrid search service application: http://blogs.msdn.com/b/spses/archive/2015/09/15/cloud-hybrid-search-service-application.aspx.

Instead of using query federation to surface results in your environment, it relies on indexing your on-premises content in Office 365. This takes away a lot of complexity setting it up, and makes it possible to mix results from SharePoint on-premises and Office 365 in a single result block. You can set up this new feature using SharePoint 2013 or SharePoint 2016.

Figure 1 shows a representation of the "old" hybrid search architecture and figure 2 shows the "new" hybrid search architecture.

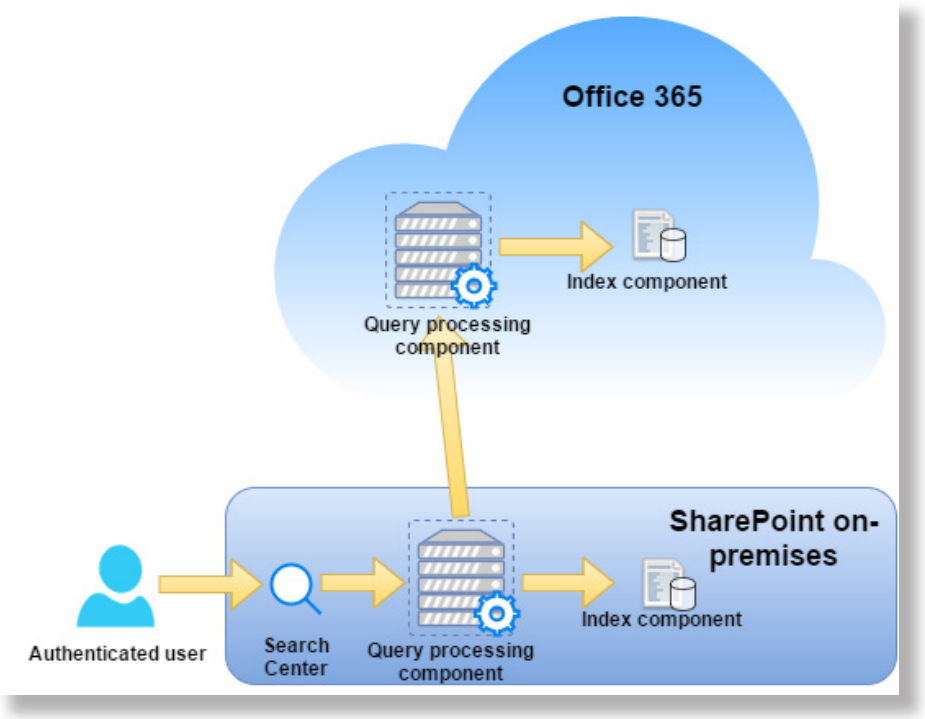


Figure 1: Hybrid federated search architecture

In this old scenario, the user enters a query in the on-premises search center. SharePoint sends the query to the on-premises query component and the SharePoint Online query component. Results cannot be interleaved out-of-the-box in this scenario, as there are separated indexes for SharePoint on-premises and SharePoint Online. However, there are several third-party solutions available that make it possible.

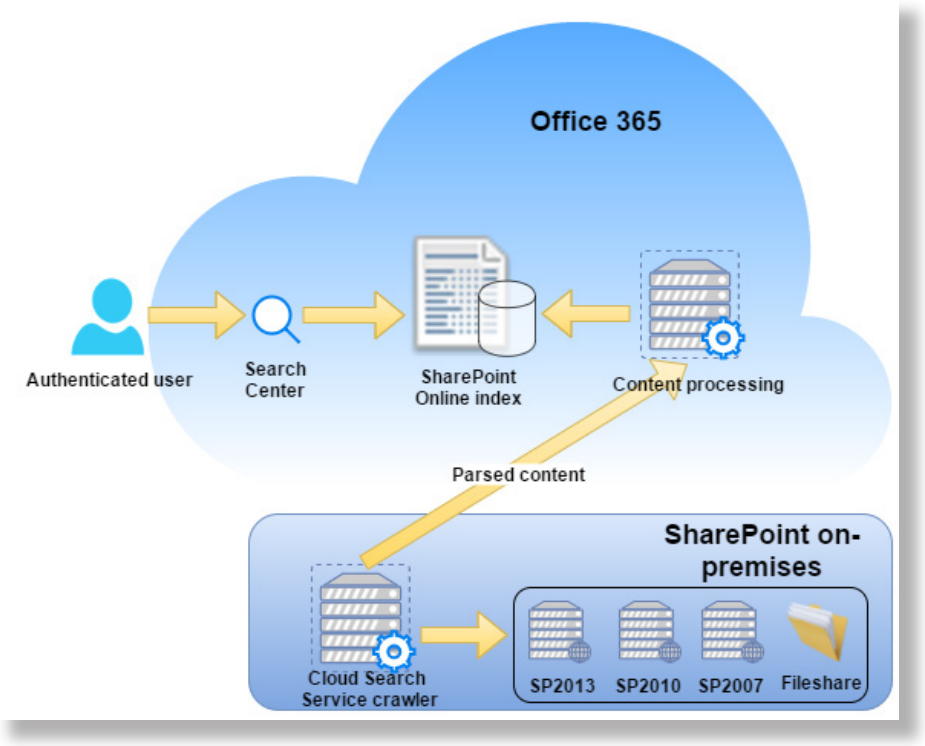


Figure 2: New cloud hybrid search architecture

In this scenario, the cloud search service crawls the content sources on-premises and sends the parsed content to the SharePoint Online content processing component. After processing the content and doing ACL mapping – for security trimming purposes – the data is saved in the SharePoint Online index. Because the index is saved online, it is now possible to interleave results in your search results and use the data in Delve.

If you want your on-premises SharePoint to show SharePoint Online results, you have to configure a remote result source in your SharePoint on-premises server. This makes sure your on-premises farms uses the SharePoint Online index.

Why cloud hybrid search

Not all companies are ready to make the move to the cloud for all their workloads. In order to help customers make the move for specific workloads, Microsoft now provides an easy way to gradually move to the cloud while maintaining a great search experience for end users.

By using the new Cloud hybrid search solution, users are able to search content from the following sources from within SharePoint Online:

- SharePoint 2007/2010/2013/2016
- File shares
- BCS

The index for all these sources is indexed in Office 365, which gives Microsoft the ability to interleave results across sources based on relevancy, use the Office 365 ranking model and even include all of this in Delve!

Organizations can also scale down search infrastructure as content processing and analytics are handled by Office 365.

Prerequisites for cloud hybrid search

In order to use the new Hybrid search functionality, make sure you have installed the following prerequisites for your environment.

SharePoint on-premises

- If you use SharePoint 2013, make sure you installed the August 2015 CU or later. I recommend the latest CU without known regressions, as there have been improvements to the hybrid search.
- Public preview of SharePoint 2016 IT Preview.

Office 365

All users that want to benefit from these new hybrid capabilities need to be licensed in Office 365.

Account synchronization

Accounts need to be synchronized to Office 365 in order to have a single identity for users.

The tools below are supported to perform directory synchronization:

- Dirsync: https://msdn.microsoft.com/en-us/library/azure/jj151831.aspx
- AADSync: https://msdn.microsoft.com/en-us/library/azure/dn790204.aspx
- AADConnect: https://azure.microsoft.com/en-us/documentation/articles/active-directory-aadconnect/

If you do not have any of the above synchronization tools deployed in your environment, I recommend using AADConnect. AADConnect can configure the full Single-sign on experience for you just by specifying server names on which it should deploy ADFS.

Software needed during configuration of hybrid search

On the SharePoint server where you are performing the configuration of hybrid search, you will need to install the following prerequisite software in this specific order.

- Microsoft online sign in assistant: http://www.microsoft.com/en-us/download/details.aspx?id=39267
- Microsoft Azure AD PowerShell: http://go.microsoft.com/fwlink/p/?linkid=236297

Onboarding script

The onboarding script will create the trust between your on-premises SharePoint environment and Office 365. You can download the script along with documentation from the Microsoft Download Center.

The scripts can be found here: https://www.microsoft.com/en-us/download/details.aspx?id=51490

Make sure you are using the latest version prior to execution.

Creating the cloud search service application

After you have installed all the prerequisites, it's now time to create the cloud search service application, which is pretty straightforward. You could use any script that you prefer; just add the parameter, "CloudIndex $true" to the New-SPEnterpriseSearchServiceApplication cmdlet.

On the server that is running SharePoint Server 2013 or SharePoint Server 2016 Preview, copy the sample script from: http://blogs.msdn.com/b/spses/archive/2015/09/15/cloud-hybrid-search-service-application.aspx and save it as CreateCloudSSA.ps1 and run it. This will create a single-server Search Service Application topology. If you want a highly available search service infrastructure, you have to manually adjust the script to your needs.

The output should look similar to figure 3.

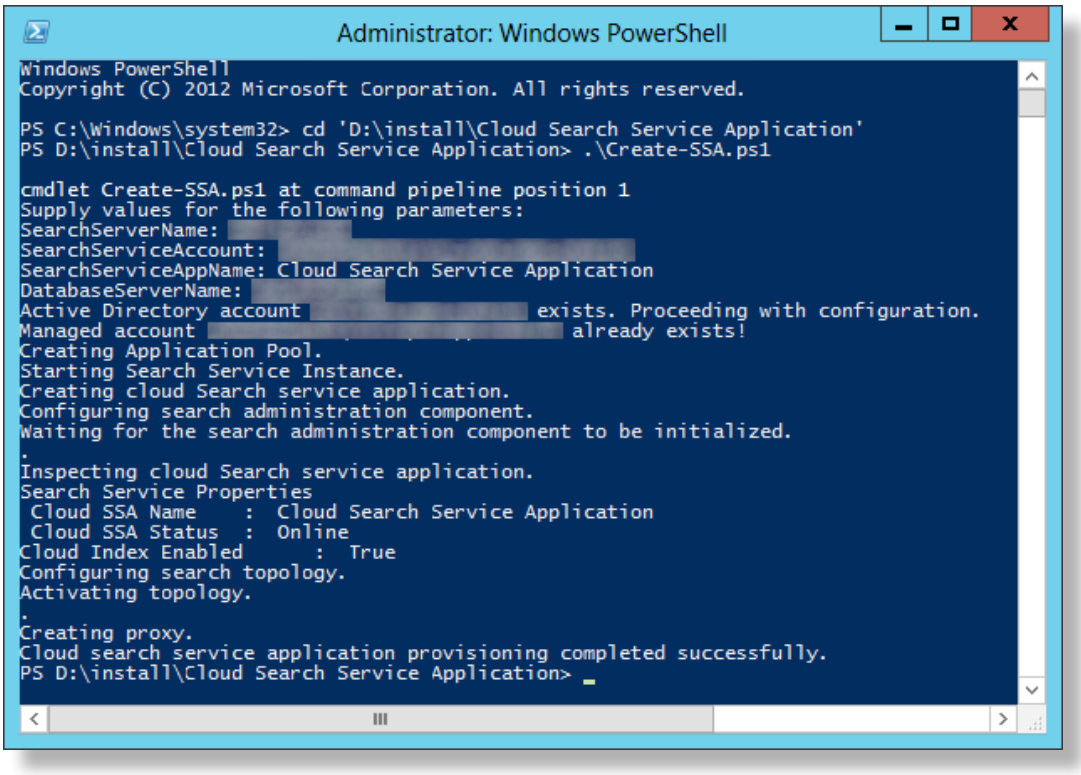


Figure 3: Create-SSA.ps1 output, creating a cloud search service application

Proxy configuration for hybrid cloud search

If your organization uses a proxy to allow Internet access, you have to configure this proxy for hybrid cloud search as well. For a more in-depth article, please look at http://sharepointrelated.com/2015/12/11/cloud-hybrid-search-proxy-settings/, but for now we can just add the proxy settings to the machine config: "C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\machine.config"

```
  <system.net>
  <defaultProxy>
    <proxy usesystemdefault="false"
     proxyaddress="http://10.1.10.1:8080"
     bypassonlocal="true" />
  </defaultProxy>
</system.net>
```

Listing 1: machine.config proxy settings

Place this anywhere between your <configuration> and </configuration> tag. To make it easier to find when you need it, you could place it right before the </configuration> tag.

Onboarding

After successfully installing the prerequisites and configuring the cloud search service application, it is time to start the onboarding process. The onboarding process will create a trust between your SharePoint on-premises and Office 365 environment. This will allow SharePoint to move the index to Office 365 for further processing.

Run the onboarding script: **.\Onboard-CloudHybridSearch.ps1** -PortalUrl "https://yourtenant.sharepoint.com" -CloudSSAId "<Cloud Search Service Application name>"

Enter your Global Administrator credentials when prompted. Figure 4 shows what your output should resemble.

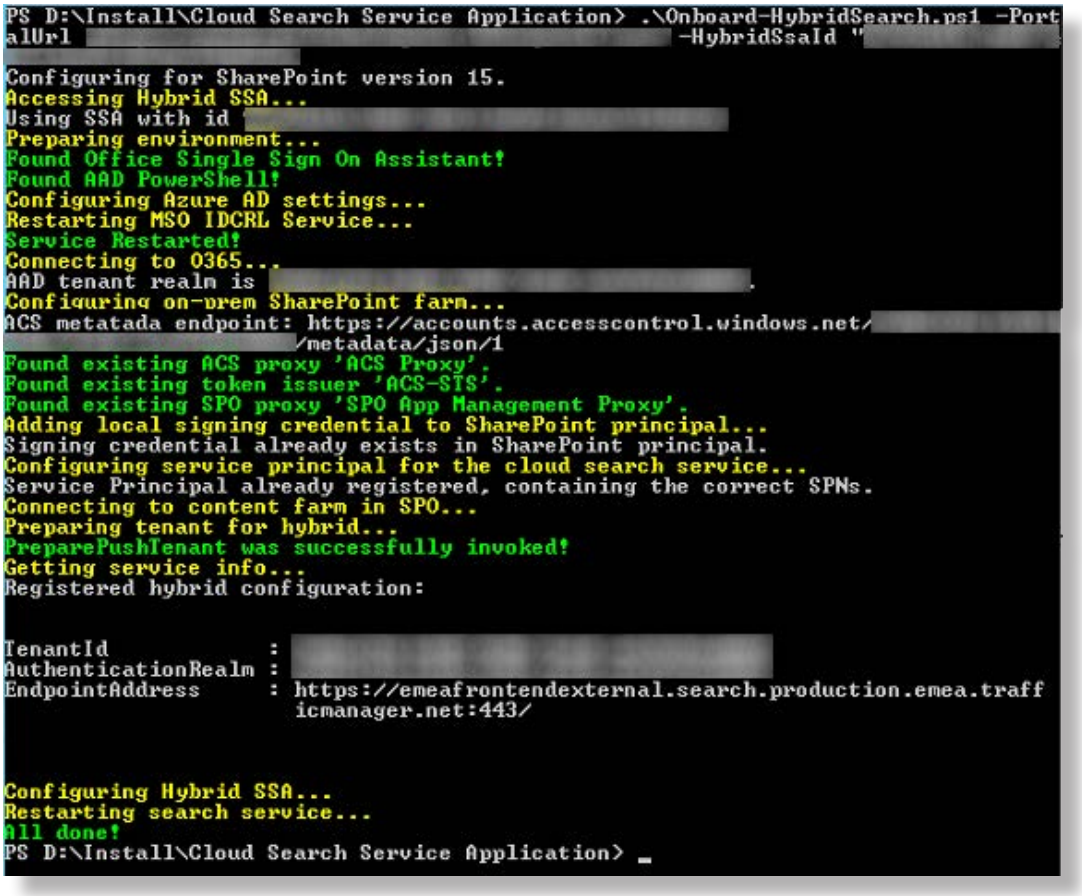


Figure 4: Running the cloud hybrid search onboarding script on the server that runs SharePoint Server 2013

The script name and the parameters have changed a bit since I ran the script. Make sure you check to see what the correct parameters are when you run the script.

Configure content source

You can configure the content source in your new cloud search service application as you would in any other on-premises SharePoint environment. As Figure 5 shows, you configure the content source of the cloud search service application in Search Administration.

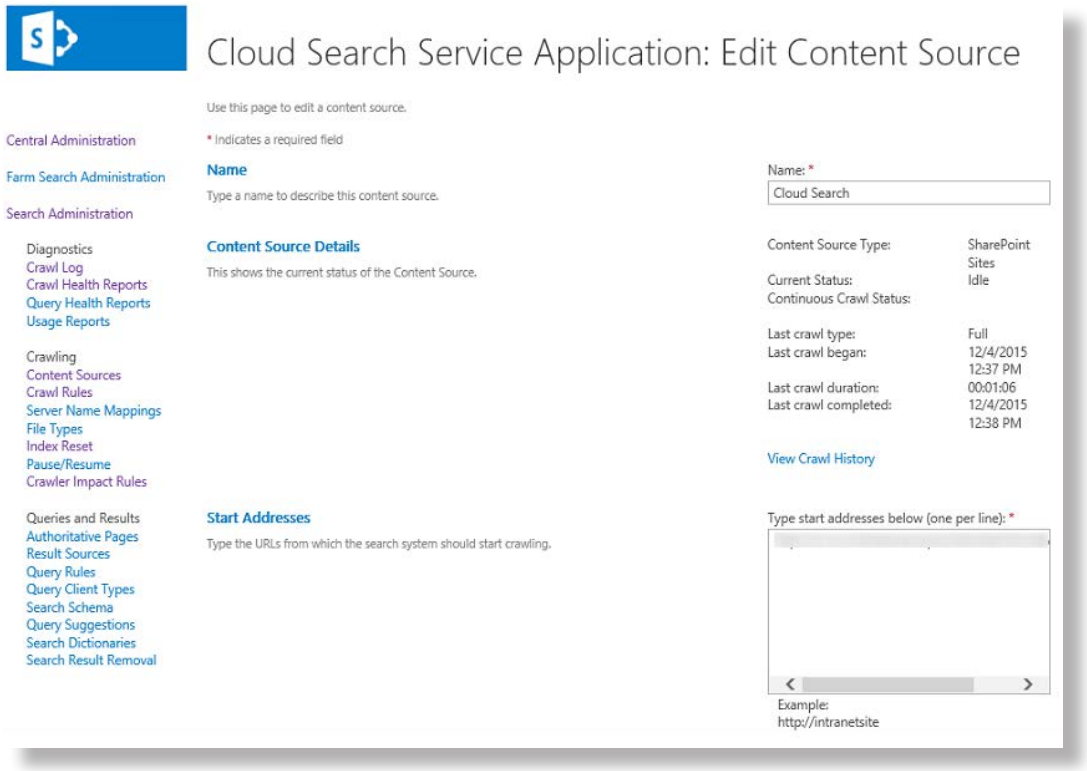


Figure 5: In SharePoint Search Administration you can edit (configure) the content source for the cloud search service application

Enter the start addresses that you would like to crawl and start a full crawl for the content source. After the crawl is done, check the crawl log for the specific content source to see if all went well.

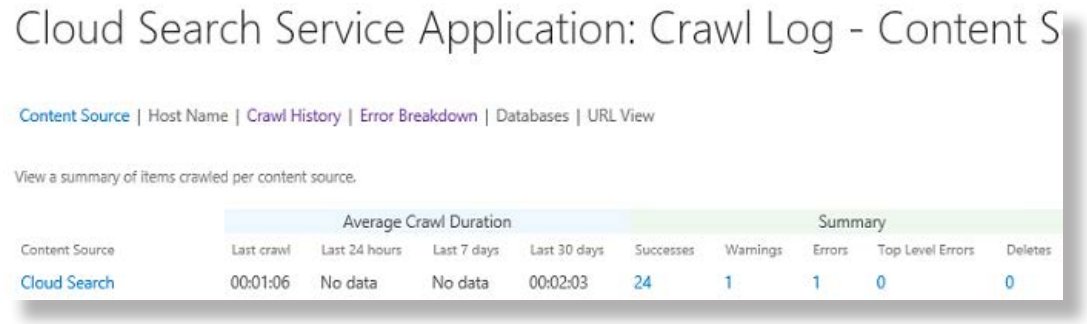


Figure 6: Check the crawl logs for any errors or warning

If you see 1 Top Lever Error with the following error message:

```
AzureServiceProxy caught Exception: *** Microsoft.Office.Server.Search.
AzureSearchService.AzureException: AzurePlugin was not able to get
Tenant Info from configuration server at Microsoft.Office.Server.Search.
AzureSearchService.AzureServiceProxy.GetAzureTenantInfo(String portalURL,
String realm, String& returnPropertyValue, String propertyName) at
Microsoft.Office.Server.Search.AzureSearchService.AzureServiceProxy.
SubmitDocuments(String azureServiceLocation, String authRealm, String
SPOServiceTenantID, String SearchContentService _ ContentFarmId,
String portalURL, String testId, String encryptionCert, Boolean
allowUnencryptedSubmit, sSubmitDocument[] documents, sDocumentResult[]&
results, sAzureRequestInfo& RequestInfo) ***
```

Listing 2: Proxy error message

Make sure to check your proxy configuration (http://sharepointrelated.com/2015/12/11/cloud-hybrid-search-proxy-settings/).

Configure your on-premises farm to use the SharePoint Online index

In order to use the SharePoint Online index in your on-premises farm, you have to configure a remote result source. This can be done by following step 4 in this article: https://technet.microsoft.com/en-us/library/dn197173.aspx.

Verifying results: perform a query in SharePoint Online and SharePoint on-premises

In Office 365 and your SharePoint on-premises farm, search for a document and it will return results for both SharePoint Online and SharePoint on-premises if cloud hybrid search is configured correctly.

Figure 7 shows example results from a search that includes the following sources:

- SharePoint Online
- SharePoint on-premises
- File shares

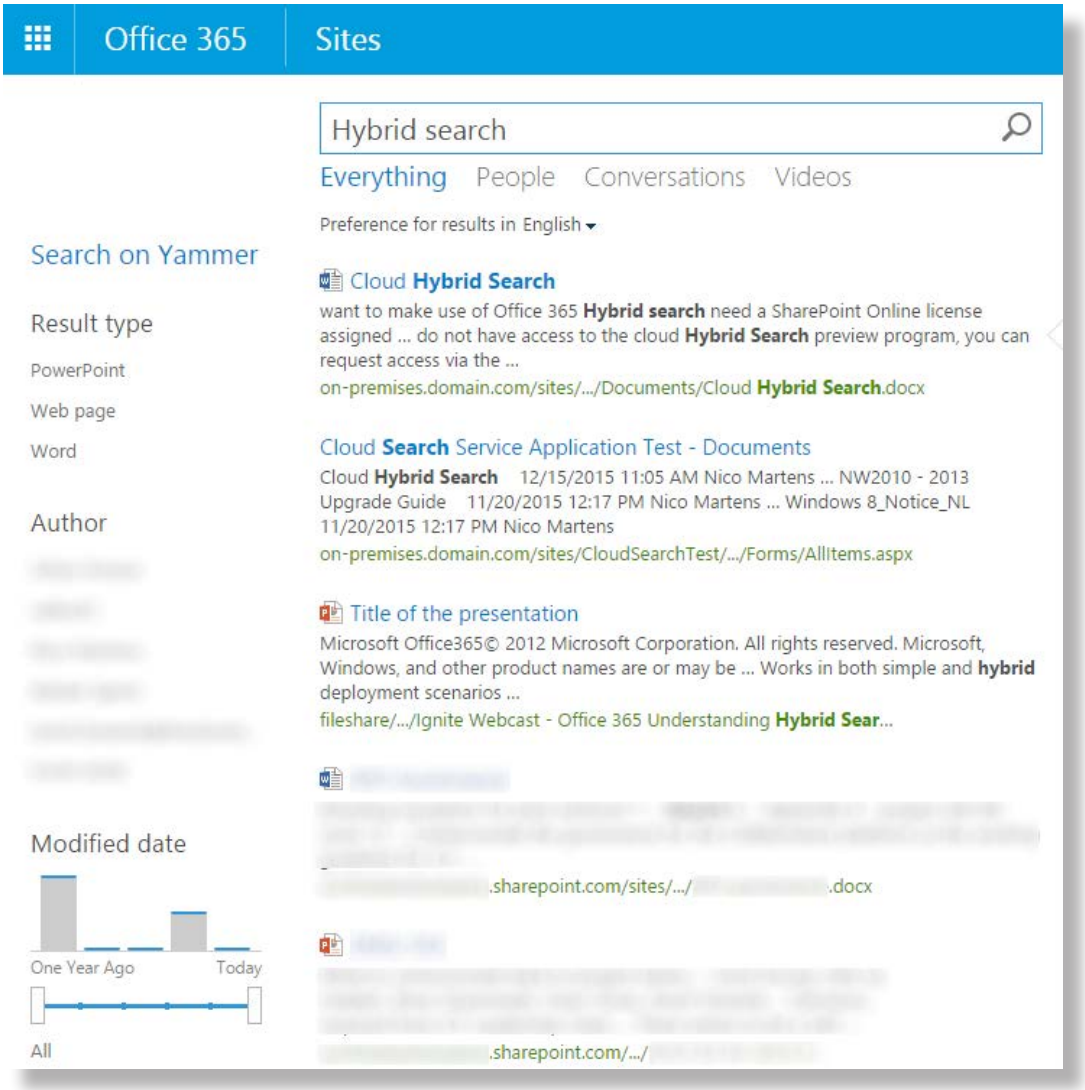


Figure 7: Searching content in Office 365 returning results from both on-premises and SharePoint Online

If you want to return results only from your on-premises site, you can use the "isexternalcontent:1" property.

As figure 8 shows, this returns only on-premises results.

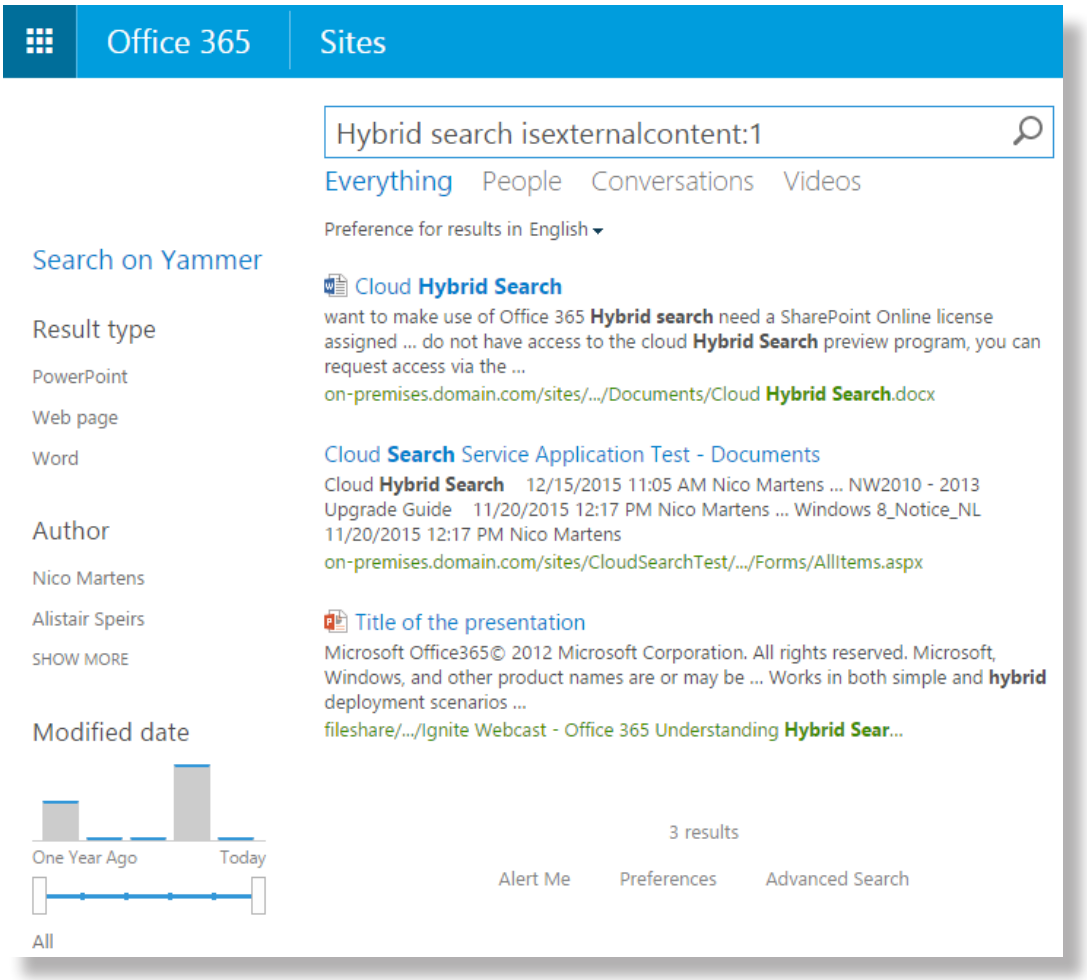


Figure 8: Using the isexternalcontent:1 property shows search results only from on-premises

That's it!

I hope this helps you use the new hybrid cloud search service application.

Conclusion

The new cloud hybrid search solution is a great way to provide a great end-user search experience, wherever the information is stored. In just a few hours, your users will be able to benefit from this new feature. If you have any problems while configuring the new cloud hybrid search capabilities, you can reach Microsoft directly by using the Cloud Search Service Application Preview forum: https://social.technet.microsoft.com/Forums/office/en-US/home?forum=CloudSSA.

Using PowerApps to create line of business solutions

by Jussi Roine

Microsoft PowerApps is a fresh take from Microsoft to tackle the persistent issue businesses have: how to easily create custom apps and solutions that use existing data and integrates with whatever backend repositories are needed? While also providing a friendly and modern interface for end users, of course.

The purpose of PowerApps is to empower the non-developers of any organization to design, create and share powerful line of business solutions. There are similarities to InfoPath, but none of the restrictions or challenges people face when building InfoPath-based forms. PowerApps is also designed from the ground up to be suitable for mobile use, as well as through a modern app on desktops.

Getting started with PowerApps

PowerApps is what Microsoft calls a gated service, thus it's currently in preview and invitation-only for organizations. It surfaces through Office 365, but can also use resources and services from Microsoft Azure through Azure API Apps (either custom or the ones that are pre-configured for PowerApps). The former is the standard way of creating and using PowerApps, the latter is used when creating apps for enterprise needs with custom integration solutions.

Figure 1: PowerApps logo

With PowerApps companies can create apps that can

- automate existing tasks
- access existing data repositories for reading and writing
- share and capture new data, such as surveys
- solve problems that typically require custom development

To start creating apps PowerApps you can sign up for an invitation here. After you've received your invitation by email, download PowerApps for your Windows desktop from Windows Store (link). Apps that you create can be accessed on iOS and Windows devices (Android support is coming later). Any app created with PowerApps is immediately accessible throughout the organization for anyone who has an iOS or Windows-based tablet or phone.

An Office 365 subscription is required for sharing PowerApps with other users.

Creating your first app with PowerApps

After installing PowerApps from Windows Store, you start by choosing your target platform, either phone or tablet.

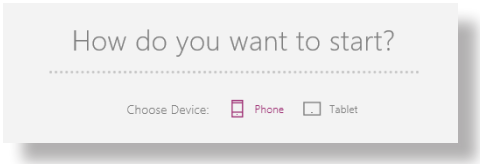


Figure 2: Choosing the target device

You can also choose between landscape and portrait orientation later in the app settings.

Your app can be based on a pre-defined template or it can be based on existing data (such as an Excel sheet). If you already have a strong vision for your app you can start with a blank app.

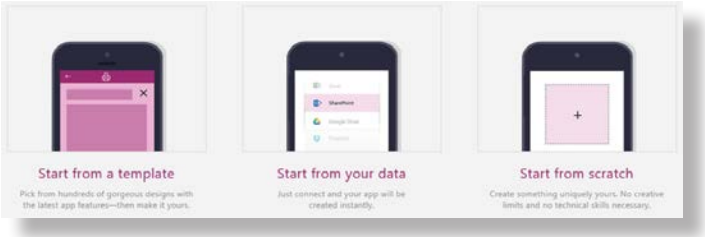


Figure 3: Starting point for creating a new app

The easiest way to is to start with a template, which there are several to choose from: conference agenda, event signup, product catalog, survey etc.

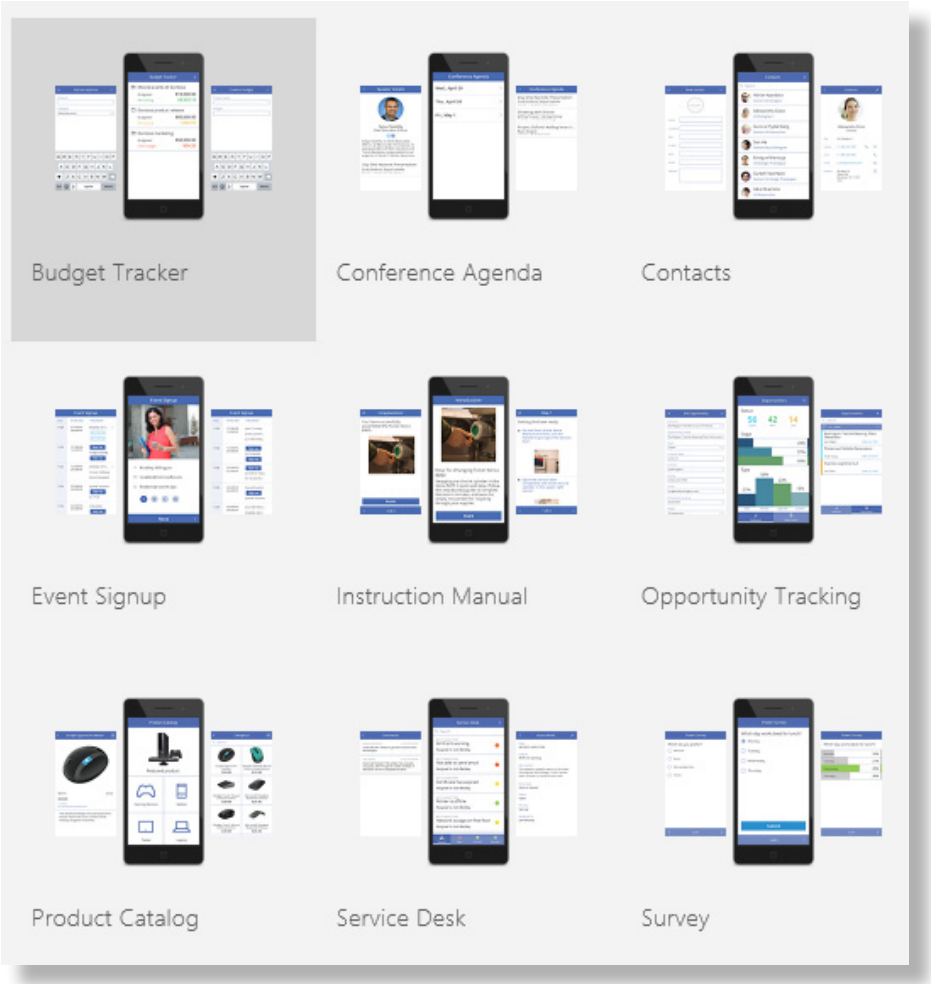


Figure 4: Pre-defined templates for PowerApps

I'm often organizing local community events so I'll choose event signup as the template as that's probably something I could benefit from immediately. Templates typically have sample data in a pre-defined Excel file, which can be stored to a selection of repositories such as SharePoint Online, Dropbox, Google Drive and OneDrive. I'm using Dropbox, so PowerApps authenticates with my credentials to Dropbox and creates a directory structure.

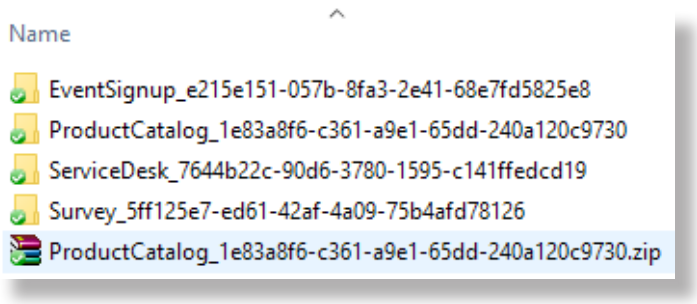


Figure 5: Data stored by PowerApps to a Dropbox directory structure

The Excel file has two sheets, one for volunteers and another for schedule.

	A	B	C	D	E
1	Name	Email	PreferredTShirtSize	__Pow	
2	Jack Creasey	jackc@contoso.com	M	Kk_gDYCplzg	
3	David Daniels	davidd@contoso.com	S	4qhCGnUYrDc	
4	Jose De Oliveira	josedo@contoso.com	S	O43aZr8JOWo	
5	Ed Dudenhoefer	edd@contoso.com	S	BEuiIxxcpc8	
6	Susan Eaton	susane@contoso.com	L	EKZHIMdSI_E	
7	Janeth Esteves	janethe@contoso.com	L	dkx62CdFfzM	
8	Jeffrey L. Ford	jeffreyf@contoso.com	S	33oWQIC3ewM	

Figure 6: Volunteers data in the sample Excel file

	A	B	C	D	E	F
1	Day	Time	Volunteer1	Volunteer2	Backup	__Pow
2	7.29	10:30AM - 12:00PM	Jack Creasey	Jenny Gottfried	Julia Moseley	4R4r4mWMm
3	7.29	12:00PM - 01:00PM		Jon Grande		yI2kZwE3oRC
4	7.29	01:00PM - 02:00PM		Jonathan Haas	Ed Dudenhoefer	SdkYf7r-NKc
5	7.29	02:00PM - 03:00PM	David Daniels		Janeth Esteves	_UyyWTMPN
6	7.29	03:00PM - 04:00PM	Jose De Oliveira	Bradley Beck		jEx4qqYNdH4
7	7.29	04:00PM - 05:00PM	Ed Dudenhoefer	Mason Bendixen	Laura Norman	jjwLQ7GHRzk

Figure 7: Schedule data in the sample Excel file

PowerApps shows the two pages for the sample app on the left, and I can add or remove content and controls from pages on the main content area.

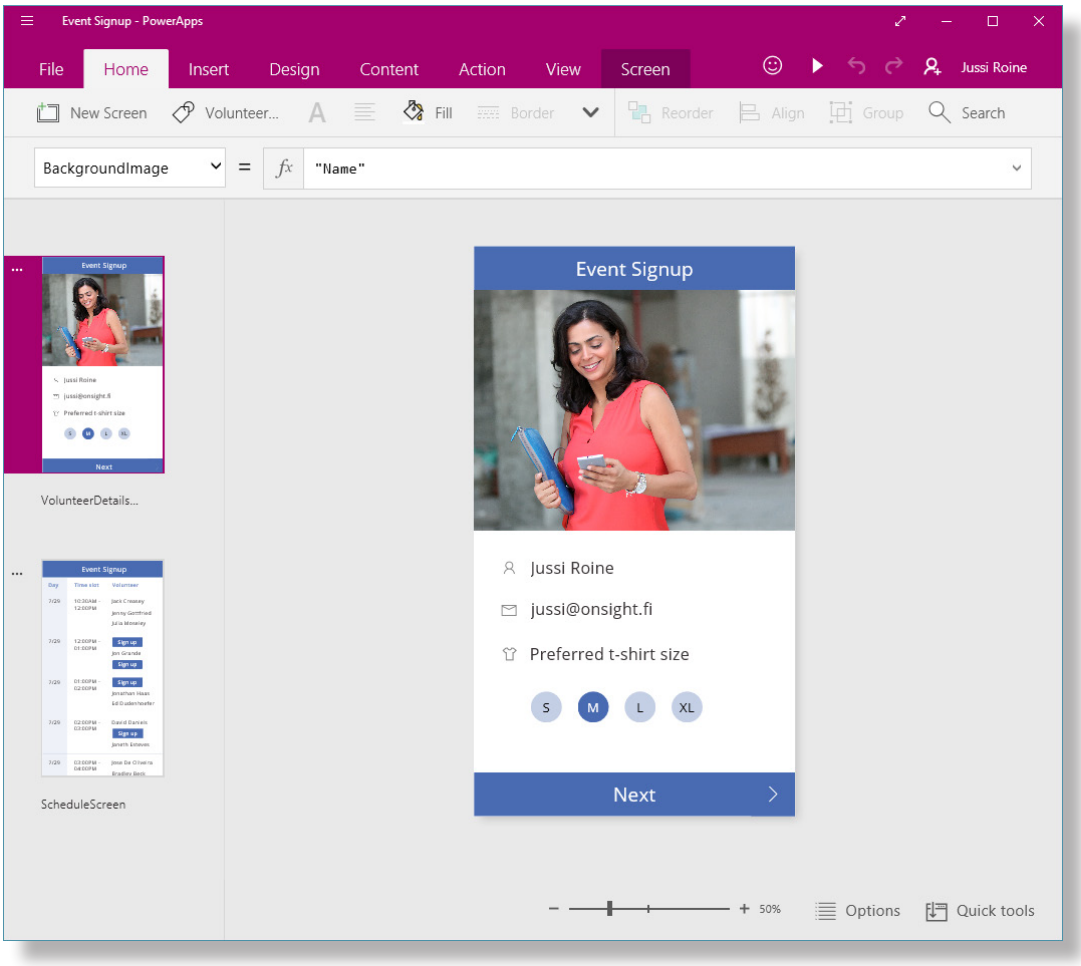


Figure 8: PowerApps design view

Running the app gives me a chance to try out signing up for an event. Selecting a t-shirt size and clicking Next gives me the schedule for signing up. Clicking Sign up adds my name on the list.

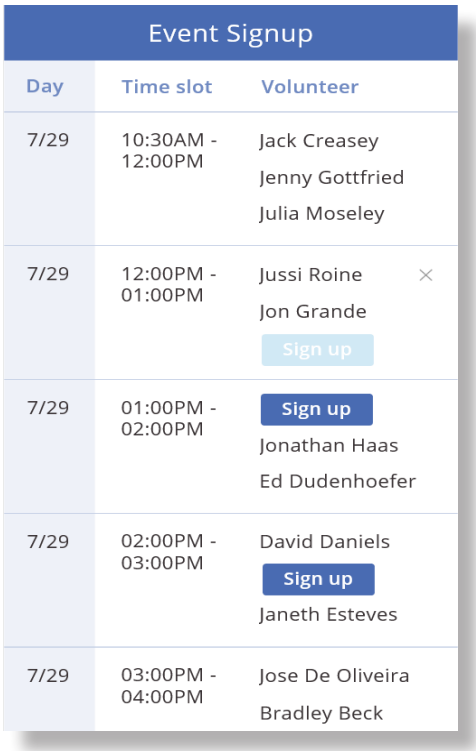


Figure 9: Using the event signup app, page 2 showing the schedule for signing up

After submitting the data, I can open the Excel again and see that PowerApps inserted the new volunteer data automatically.

	A	B	C	D
1	Day	Time	Volunteer1	Volunteer2
2	7.29	10:30AM - 12:00PM	Jack Creasey	Jenny Gottfried
3	7.29	12:00PM - 01:00PM	Jussi Roine	Jon Grande
4	7.29	01:00PM - 02:00PM		Jonathan Haas
5	7.29	02:00PM - 03:00PM	David Daniels	
6	7.29	03:00PM - 04:00PM	Jose De Oliveira	Bradley Beck

Figure 10: New data has been added to the Excel after using the app to sign up

In just few clicks I was able to create a sample app, integrate that with Dropbox for data storage and test the app!

Sharing the app

Now that my app is ready I can share it with other users. I can share it through Office 365 or save it locally. The app is saved as an .msapp file, that can be distributed as needed.

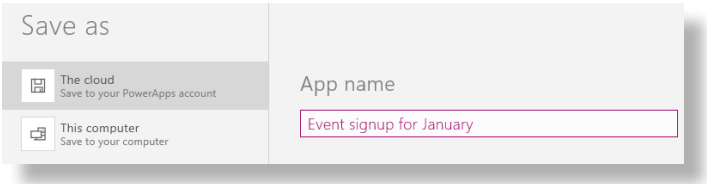


Figure 11: Sharing the app to the cloud (Office 365)

Opening a browser and navigating to https://web.powerapps.com I can now view the app and share it with desired recipients. The app is only accessible with users who have been invited to use the specified app. I can designate view or edit permissions for each recipient separately.

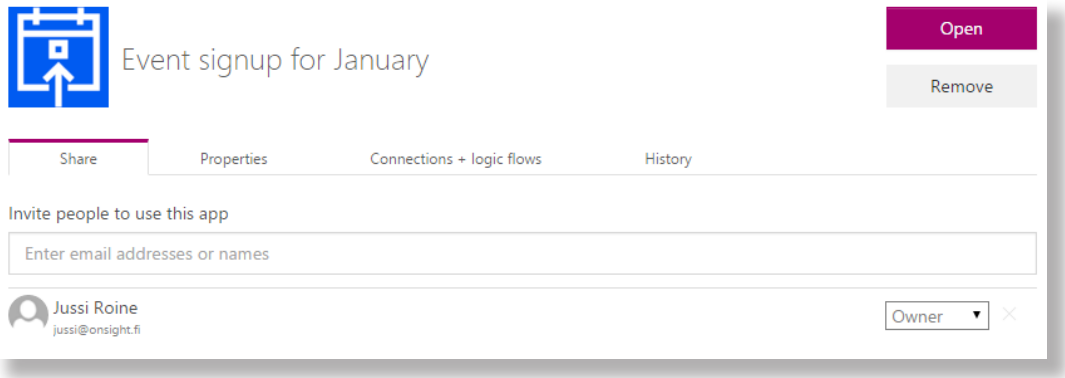


Figure 12: Managing the published app from the web

Creating the user interface

Working with the user interface within your app is fairly easy. You can embed controls, such as text boxes, checkboxes, buttons and sliders on a view within the app.

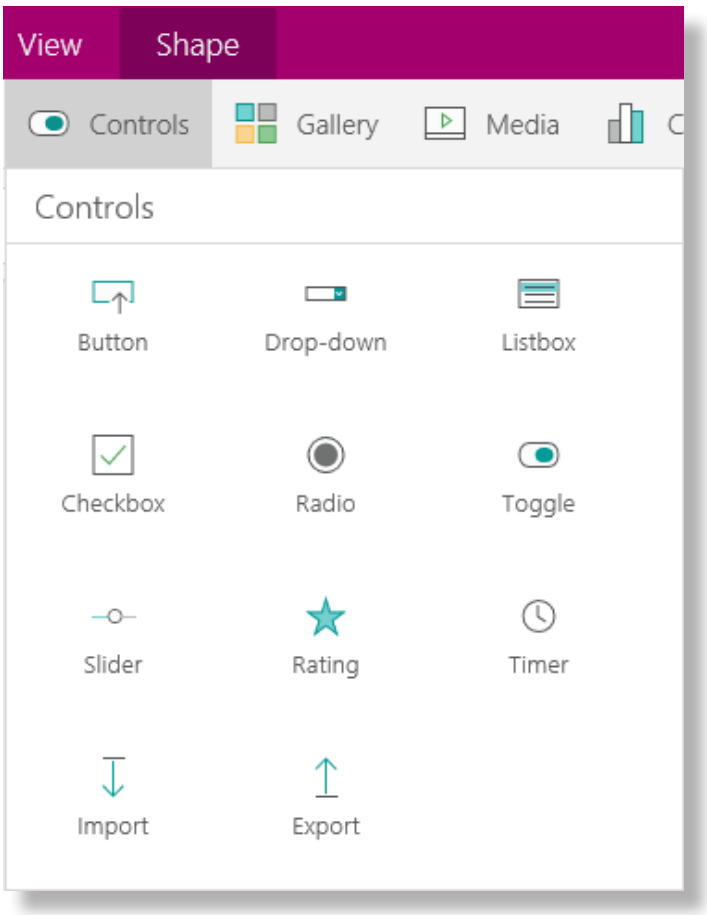


Figure 13: Shaping the app with controls

You can also use predefined layouts, colors and themes to quickly create new pages within your app. Audio, video and other media can be embedded for rich content on pages. Functionality and logic can be added with formulas, which are a lot like Excel-based formulas. By combining controls, formulas and connections to backend data, most decisions for navigating the app and reacting to user inputs can be easily done.

Advanced uses for PowerApps

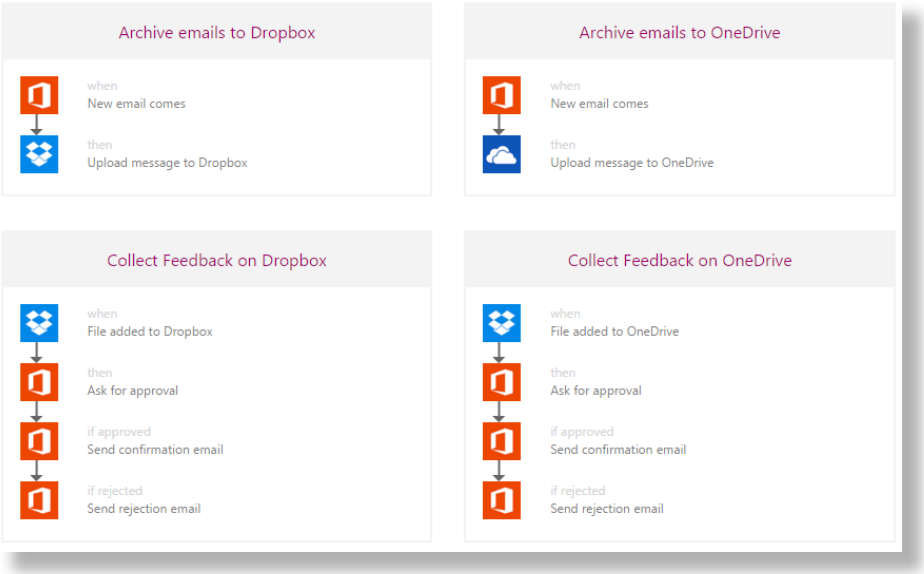


Figure 14: Pre-defined logic flow templates

Besides simple forms that can be filled and data that can be captured and stored, PowerApps has much more to offer. One of the more important features is Logic Flows, that allow app designers to create logical flows based on triggers from external services. This feature can be accessed through PowerApps.com, and is also based on templates.

I'll create a logic flow that checks for updates in my previously created event signup app. The Excel file for volunteers sits in my Dropbox, so I can now check for any changes to the Excel file. Interestingly logic flows can be created without the Windows app for PowerApps using just a browser.

My logic flow looks like this after selecting the file from Dropbox, and configuring the notification email properties.

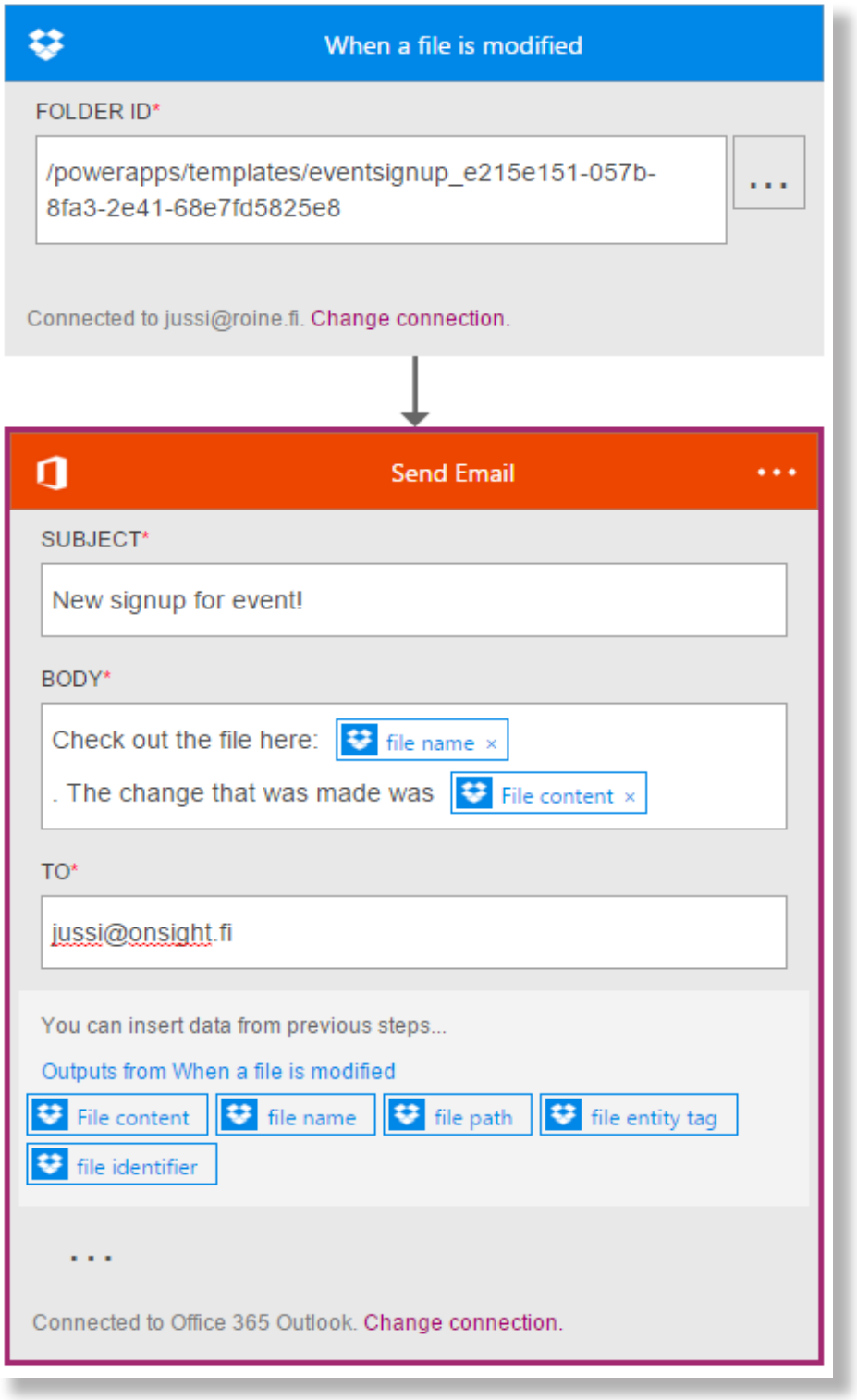


Figure 15: Logic flow steps with configuration in place

Opening each block of logic, I can fine tune what is being done and how. I can also add new steps or conditions to the logic flow.

Saving the logic flow automatically activates it. I then went back to my event signup app, and enrolled myself for another time slot. This triggers the logic flow, and sends me an email with the details.

Since PowerApps is in preview I experienced random timeouts and issues with logic flows. For example, even though I had authenticated successfully against Exchange Online, the Send_Email would sometimes just error out.

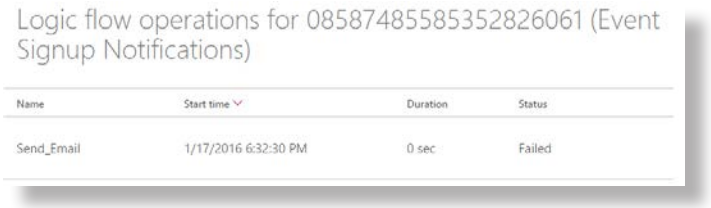


Figure 16: Logic flow troubleshooting is very limited at the moment

I'm hopeful these basic issues will be resolved during the preview of PowerApps, or at least a better diagnostics mechanism is established.

PowerApps plans and licensing

PowerApps is free of charge for basic use. The free plan allows users to create and use an unlimited number of apps, but is restricted to 2 connections to SaaS data sources per user. All apps also run on a shared infrastructure.

The paid tier is called Standard, but pricing for this plan has not yet been revealed. It is free during the preview period. The difference in Standard tier is unlimited connections to SaaS data sources, but other than that it's the same as the free offering.

PowerApps and Office 365

When PowerApps is activated for the Office 365 tenant, it becomes visible in the app launcher just like any other Office 365 service.

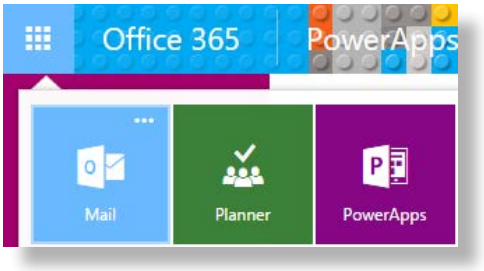


Figure 17: Accessing PowerApps in Office 365 navigation

Conclusion

While PowerApps is still very new and companies are starting to try it out, it already looks very promising. I see PowerApps as a great platform for easily creating custom apps for running your business. It provides a great way for users to access apps without a steep learning curve, and sharing apps through Office 365 is very easy.

If there's a need for more complex solutions, PowerApps can still act as a frontend and the heavy lifting can be provided through Azure API Apps and the wealth of services from both Office 365 and Microsoft Azure.

Some might say PowerApps is what Infopath was promised to be but never was. While native integration with SharePoint and PowerApps is (for now, at least) very limited, I can't say I'm missing a minute from InfoPath, and already love the way I can rapidly start building solutions for customers with PowerApps.

About the Authors

Introducing OfficeDev Patterns and Practices

Paolo is a consultant, trainer, book author and speaker at the best international conferences about Microsoft technologies. He works in a company of his own, called PiaSys.com, and he is focused on Microsoft Office 365 and Microsoft SharePoint.
He writes articles for IT magazines, authored several books for Microsoft Press and posts regularly on his technical blog (http://www.sharepoint-reference.com/blog) and Twitter (@PaoloPia).
Paolo passed about 50 Microsoft certification exams including the Microsoft Certified Solutions Master - Charter SharePoint. In 2015 was awarded as MVP on Microsoft Office 365.
Since 2015 he is member of the Core Team of the Office 365 Developer Pattern and Practices project.

Information management Office 365

Albert Hoitingh is business consultant working for the Microsoft business unit at Sogeti Netherlands. He's been working with information worker solutions from the late 1990's, starting with IBM Lotus Notes. He has extensive experience with Microsoft SharePoint (2001-2013, Online) as-well-as other Microsoft platforms. Albert enjoys inspiring organizations as a SharePoint evangelist and works closely with business users to successfully implement SharePoint solutions. He's presented at SharePoint Connections in Seattle and in Amsterdam as well as the Sogeti SharePoint events. When he's got the time, he blogs at alberthoitingh.wordpress.com.

The Insider Security Threat, and How SharePoint / Office 365 Measures Up

Peter Bradley has been a SharePoint professional since 2002. He has been a developer, business analyst, project manager, and architect on hundreds of SharePoint projects large and small, in Australia, Switzerland, the US and UK. He has specialized in secure information management for many years, and in 2014 he founded Torsion Information Security.
Peter hails from Perth, Western Australia, and has called London home for 7 years. He enjoys spending as much time with friends, travelling, and playing football, drums and guitar as he can find.
email: peter.bradley@torsionis.com
blog: www.torsionis.com/blog
t: @torsionis

A new world of Front-End development

Hi my name is Cas van Iersel from Point 42 and I'm an experienced SharePoint Developer with love for Design and Interaction. Working with SharePoint since early Beta 2007, now developing all kinds of Solutions, Features and Apps for SharePoint 2010 / 2013 / Office 365

Office 365 collaboration tools: what to use for what?

Frédérique Harmsze works as a consultant at Macaw. She has been doing SharePoint projects since 2004, both in small organizations and in international teams for multinationals. In her work, she focuses on the client and the end-user, rather than the technical aspects. She plays a role in the implementation, from the functional angle, of SharePoint and Office 365 and is often involved in user adoption. She blogs about it on her website http://frederique.harmsze.nl

Practical guidance for deploying high trust add-ins on-premises

Andries den Haan works as an engagement architect for ETTU.
I've been involved in various roles and projects using portal and collaboration technologies since 2000.
My expertise is mainly focused around leveraging web technologies the right way to build valuable business solutions within the digital workplace, including corporate portals, social networking platform, collaboration workspaces, enterprise search and content management.
To build a true digital workplace, one needs to embrace the long term view and commit to truly transforming business collaboration!
Contact me via a.haan@ettu.nl or via LinkedIn.

Developing Office 365 apps with the Microsoft Graph and Office UI Fabric

Maarten Stelling is a SharePoint and Office 365 enthusiast, living in Utrecht and working as a SharePoint consultant at Rubicon. Maarten has a passion for developing web applications with a main focus on SharePoint and Office 365 development. He is a MCSD SharePoint Applications and always looking for more certificates on the Microsoft stack. Maarten tries to contribute to the community by writing articles and by blogging on http://www.maartenstelling.nl.
Twitter: @stellingm

Welcome back old friend

Jasper Oosterveld is a frequently requested speaker and co-author of two SharePoint books. As a Cloud Consultant at Sparked, he facilitates seamless implementations of SharePoint Online and Office 365. As a consultant, Jasper knows how to guard the bottom line as well as operate at the detail level. He works flexibly and convincingly towards an optimum result for his clients.
Jasper is the "Share" in SharePoint. As a co-founder of the Dutch SP&C site, he supports business users of Office 365 and SharePoint by providing the latest information daily. He has received honors these activities: The Microsoft MVP awards since 2014. Deservedly, Jasper can be named the "go-to guy" for SharePoint and Office 365.

Everything you need to know about cloud hybrid search

Nico Martens is a SharePoint technical consultant at Portiva, one of the larger SharePoint and Office 365 implementation partners in the Netherlands. On a daily basis, he helps organizations implement SharePoint / Office 365.
Nico likes to work in complex scenarios, working with the newest technologies available to deliver the best experience for the customer. Areas of expertise are SharePoint, Office 365, identity management and hybrid solutions. As an IT Pro he loves to use PowerShell to automate tasks.
Twitter: @MartensNico
Blog: http://sharepointrelated.com
LinkedIn: http://nl.linkedin.com/in/martensnico/

Using PowerApps to create line of business solutions

Jussi Roine is Microsoft MVP, Microsoft Certified Master and MCT Regional Lead for Finland, Twitter: @jussiroine, email: jussi@onsight.fi

DIWUG
Dutch Information Worker User Group

10 YEARS AHEAD OF THE PACK

DIWUG event 26 mei 2016 bij Rubicon Vianen
Dave Koonen vertelt over testautomatisering

DIWUG event gemist? Volg #DIWUG.

Notes & Remarks

Technology

People

Fun

SharePoint

Kom werken aan uitdagende O365 projecten!

avanade®
Results Realized

Passion

Office 365

Heb jij een passie voor Microsoft-technologie en ben je toe aan een nieuwe stap in je carrière? Bekijk jouw nieuwe uitdaging op werkenbijavanade.nl en neem contact op met Duygu Ciftci of Marsha Jurgens via nl.recruitment@avanade.com of 036 547 51 07

avanade®
Results Realized

From Accenture and Microsoft