

DBXL OFFICE 365

DBXL INSTALLATION/CONFIGURATION AND UPGRADE DOCUMENTATION

VERSION 1.0

This document mentions the steps for those customers who wants to setup Dbxl on their azure environment itself.

Infrastructure Requirements:

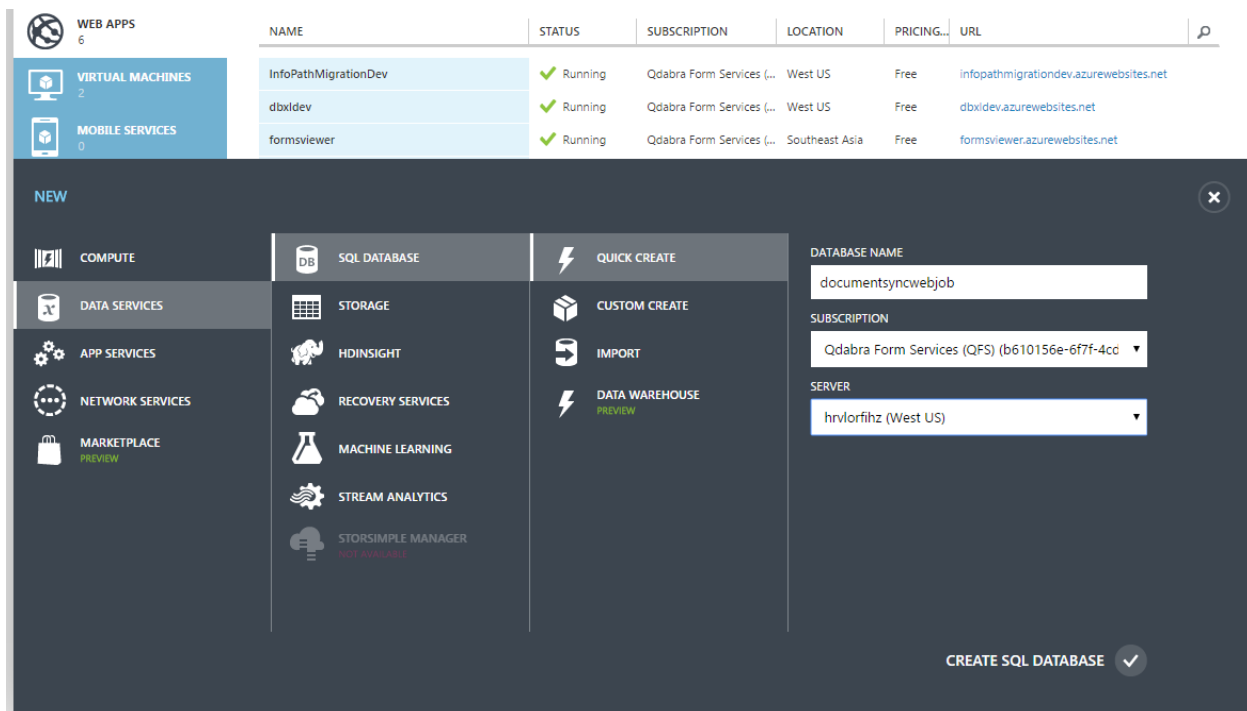
1. Office 365 Account (for App package).
2. Azure Account (for web package).
3. Hosted Qdabra DBXL web service using a static IP. So that azure hosted site can access the service using credentials. (To submit document to DBXL).
4. SQL Server Instance (To save the documents on DBXL). Actually this is the part of DBXL service installation.
5. SQL Server Instance (Azure) for saving document synchronization status.
6. Microsoft Office InfoPath for the configuration of Document Type.

Steps to setup the packages:

To setup the provider hosted app on office 365 for DBXL needs the following things to do:

Create a database on SQL Azure:

Create a database in SQL Azure by login to Azure portal type the database name, specify the server and click on Create SQL Database button.



Now connect this database in local sql server and run the .sql file (SyncStatus.sql provided with documents, or see [Appendix](#)) to create the 'SyncStatus' table. This table basically holds the information of documents synchronized.

Publish the web project or package on Azure:

We can publish our web project in azure from different source control repositories like TFS, GitHub, bit bucket, Dropbox, Visual Studio etc. In this guide we are using Visual Studio to deploy our web project.

1. Login to azure account click on New -> Compute -> Web App -> Quick Create.
2. Give it a URL and select App service plan.
3. Click on create web app.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with 'Microsoft Azure', 'CREDIT STATUS', 'Subscriptions', and a user profile. Below this, the 'all items' page is displayed, showing a table of resources. The table has columns for NAME, TYPE, STATUS, SUBSCRIPTION, and LOCATION. The resources listed are EchoService29, JimmyMachines, jimmytestserver, and dbxinstances. Below the table, there's a 'NEW' button that opens a menu with options like COMPUTE, WEB APP, and QUICK CREATE. The 'WEB APP' section is expanded, showing options like VIRTUAL MACHINE, MOBILE SERVICE, CLOUD SERVICE, and BATCH SERVICE. The 'QUICK CREATE' section shows 'CUSTOM CREATE' and 'FROM GALLERY'. On the right side of the 'NEW' menu, there's a form to create a new web app, with fields for URL (DBXLDemo.azurewebsites.net) and APP SERVICE PLAN (Default0 (North Central US, Free)). A 'CREATE WEB APP' button is at the bottom right of the menu.

NAME	TYPE	STATUS	SUBSCRIPTION	LOCATI...
EchoService29	Cloud service	✓ Created	Jimmy MSDN	Japan West
JimmyMachines	Cloud service	✓ Running	Jimmy MSDN	Japan West
jimmytestserver	Cloud service	✓ Running	Jimmy MSDN	East Asia
dbxinstances	Storage Account	✓ Online	Jimmy MSDN	North Centra...

4. Now you can see your web app under all items -> web apps.

Microsoft Azure | CREDIT STATUS | Subscriptions | himanshu.nigam@daffodilsw.com

ALL ITEMS

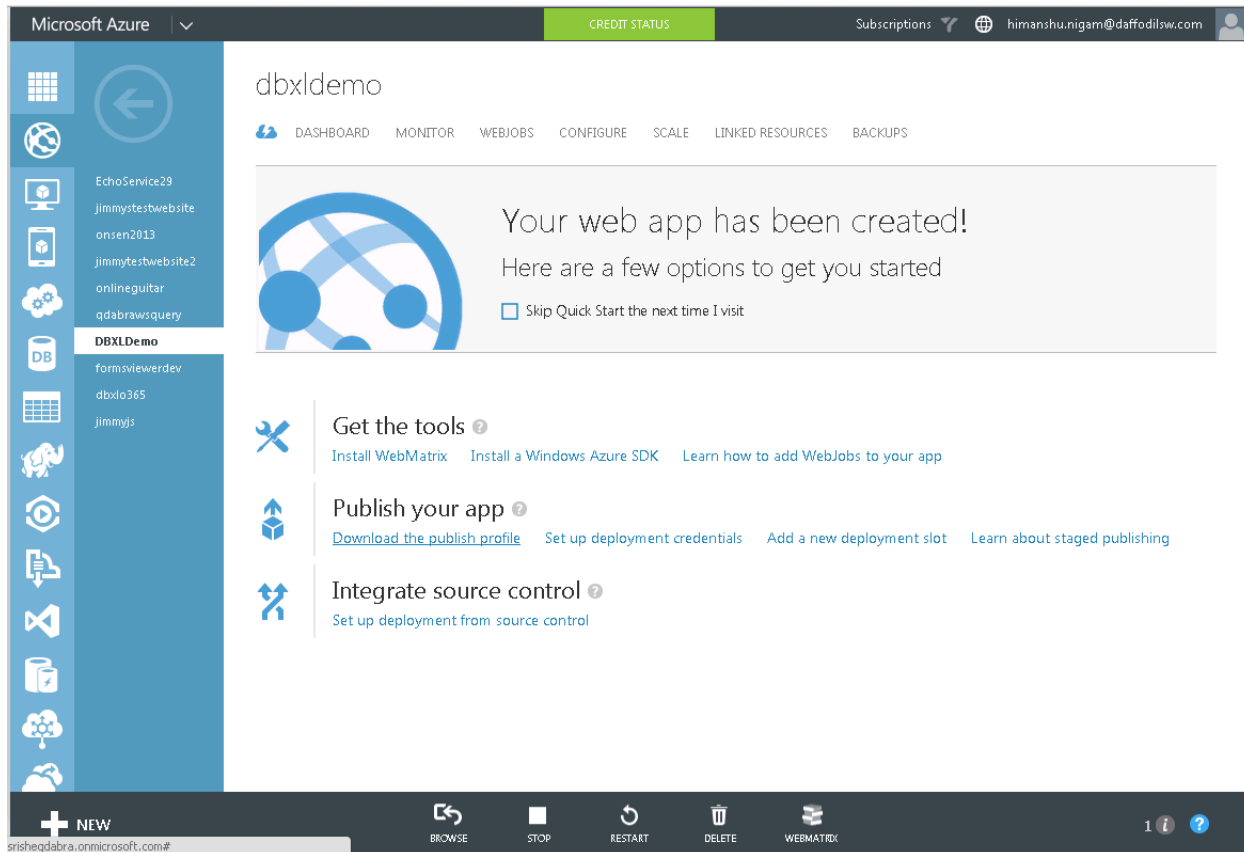
WEB APPS 10

web apps

NAME	STATUS	SUBSCRIPTION	LOCATION	PRICIN...	URL
jimmyjs	✓ Running	Jimmy MSDN	Japan West	Free	jimmyjs.azurewebsites.net
jimmytestwebsite2	✓ Running	Jimmy MSDN	North Central US	Free	jimmytestwebsite2.azurewebsites.net
onlineguitar	✓ Running	Jimmy MSDN	North Central US	Free	onlineguitar.azurewebsites.net
qdbrawsquery	✓ Running	Jimmy MSDN	North Central US	Free	qdbrawsquery.azurewebsites.net
formsviewerdev	✓ Running	Jimmy MSDN	North Central US	Free	formsviewerdev.azurewebsites.net
dbxio365	✓ Running	Jimmy MSDN	North Central US	Free	dbxio365.azurewebsites.net
EchoService29	✓ Running	Jimmy MSDN	East Asia	Free	echoservice29.azurewebsites.net
jimmystestwebsite	✓ Running	Jimmy MSDN	East Asia	Free	jimmystestwebsite.azurewebsites.net
onsen2013	✓ Running	Jimmy MSDN	East Asia	Free	onsen2013.azurewebsites.net
DBXLDemo	✓ Running	Jimmy MSDN	North Central US	Free	dbxldemo.azurewebsites.net

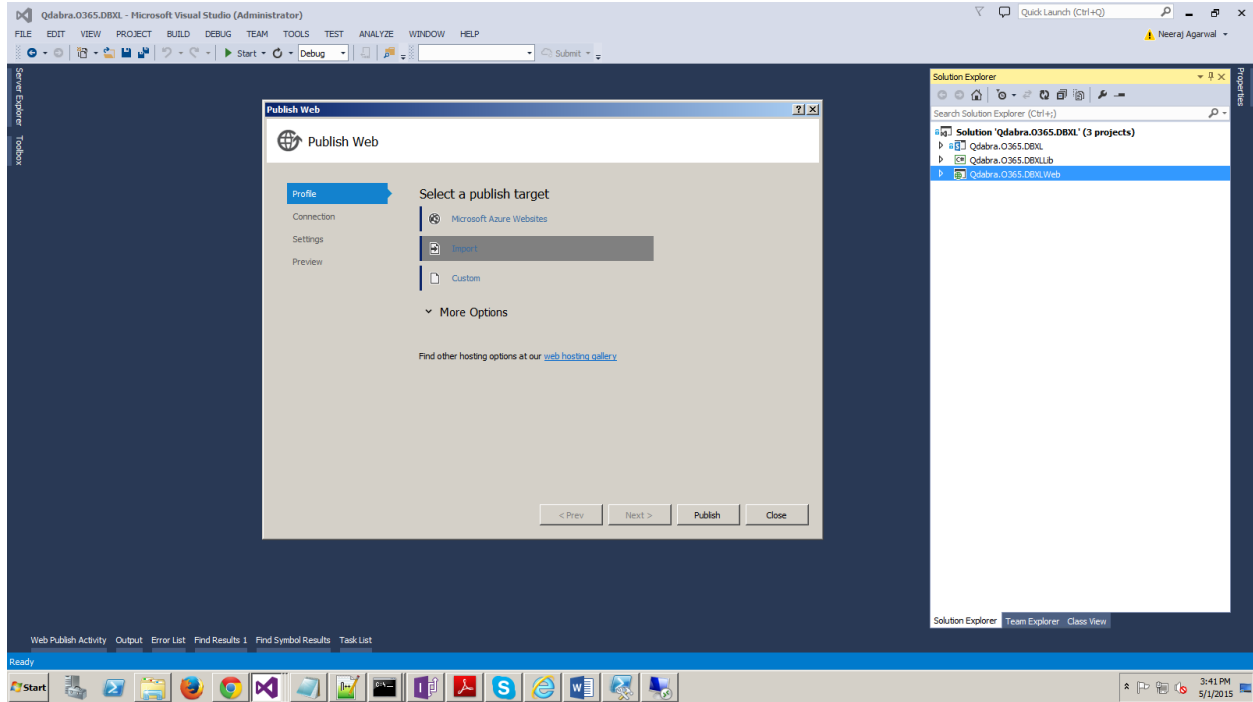
+ NEW | BROWSE | STOP | RESTART | MANAGE DOMAINS | DELETE | WEBMATRIX | 1 1 1 ?

- Now download the published profile from Publish your app link.

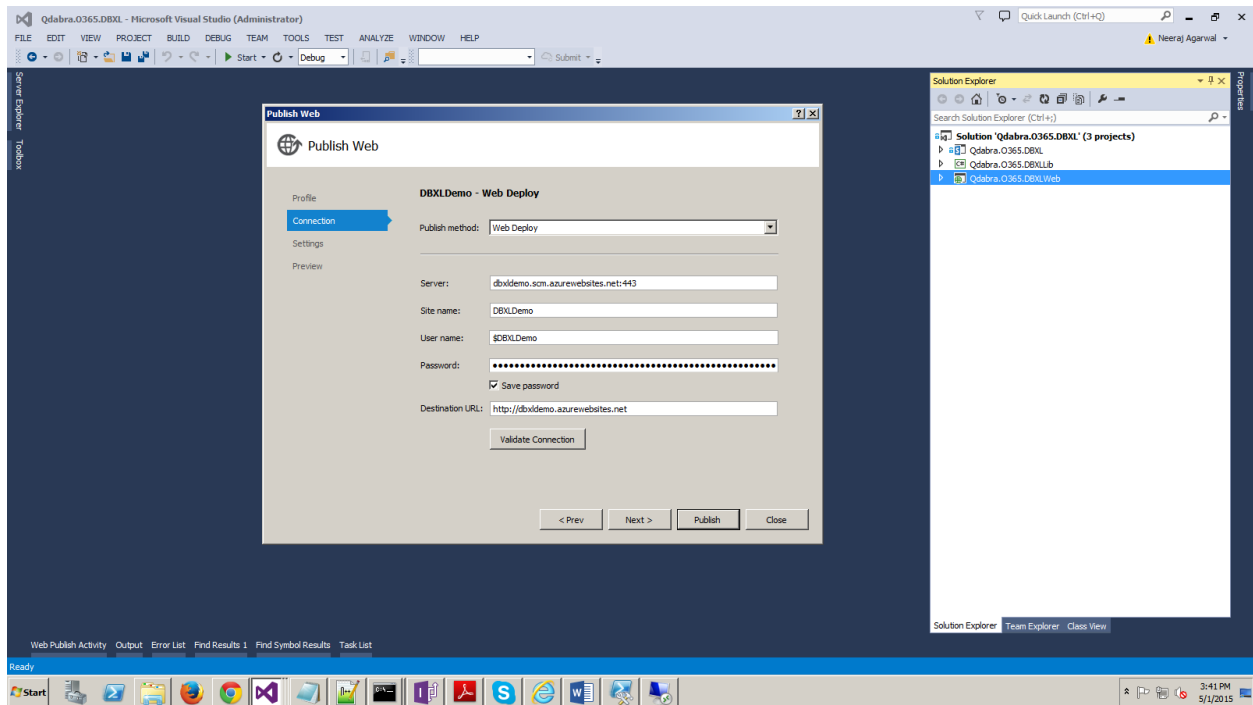


6. Now open the source code in Visual Studio 2013.
7. Right click on web project and click on publish

DBXL installation/Configuration and upgrade



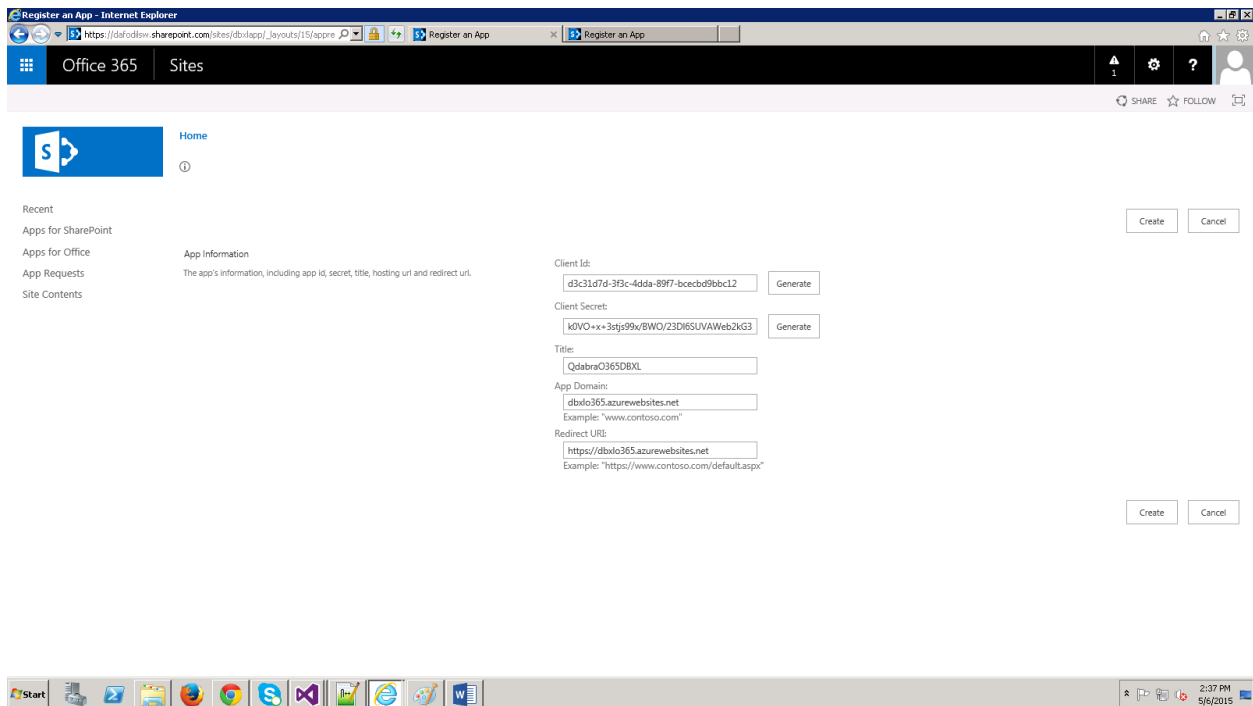
8. Click on import and select the publishing profile that we are downloaded in 5th step.



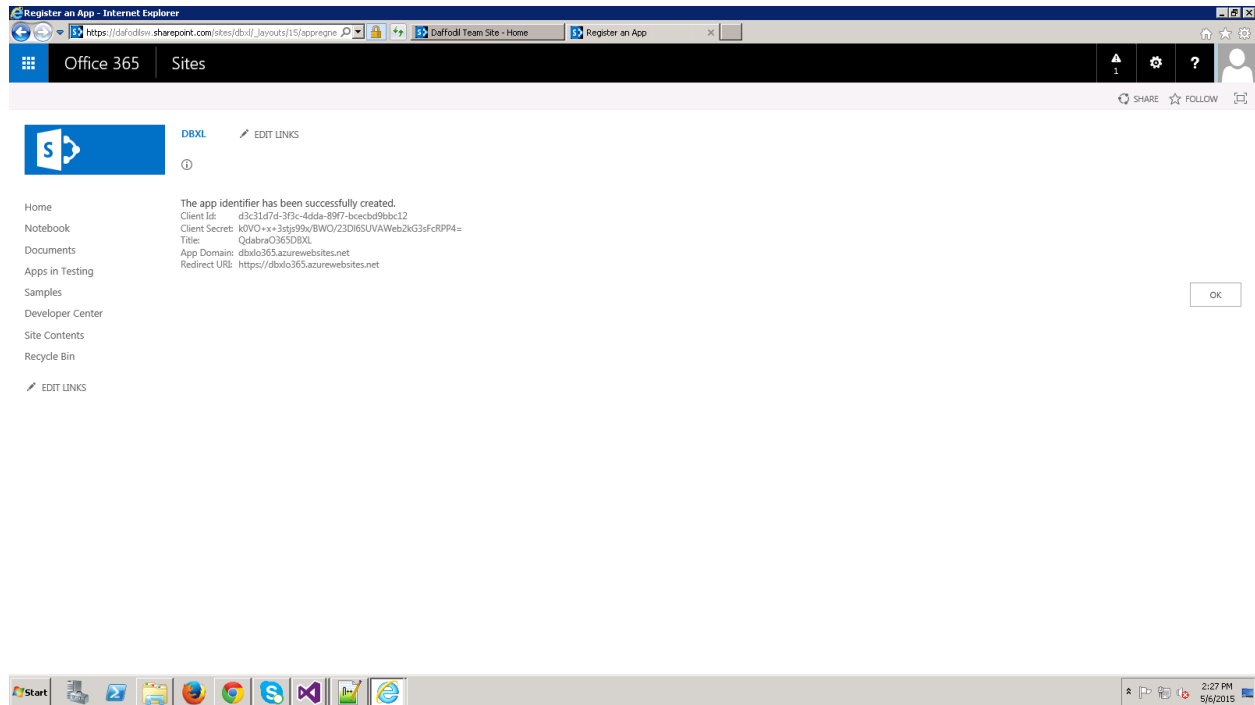
9. Click next in configuration section select Debug mode to Release if it is not.
10. Click next and publish the web project. This will publish your web project to azure.

Register Azure site with SharePoint:

1. We have to register the app with the client Id and client secret to authenticate it with SharePoint. To register it follow the steps given below:
 - a. Suppose you have a site collection with URL e.g. <https://qdabra.sharepoint.com>.
 - b. Then navigate to https://qdabra.sharepoint.com/_layouts/15/appregnew.aspx
 - c. Enter the settings on the page as mentioned below:
 - i. Client ID: Click on Generate it will give you something like this (d3c31d7d-3f3c-4dda-89f7-bcecbd9bbc12).
 - ii. Client Secret : Click on Generate it will give you something like this (k0VO+X+3stjs99x/BWO/23DI6SUVAWeb2kG3sFcRPP4=)
 - iii. Title : Title of your app e.g. QdabraO365DBXL
 - iv. App domain: dbxldemo.azurewebsites.net (Azure hosted site domain)
 - v. Redirect URL : <https://dbxldemo.azurewebsites.net>
 - d. Click on create button to save the settings.



2. Next window will show the entered details of app registration.

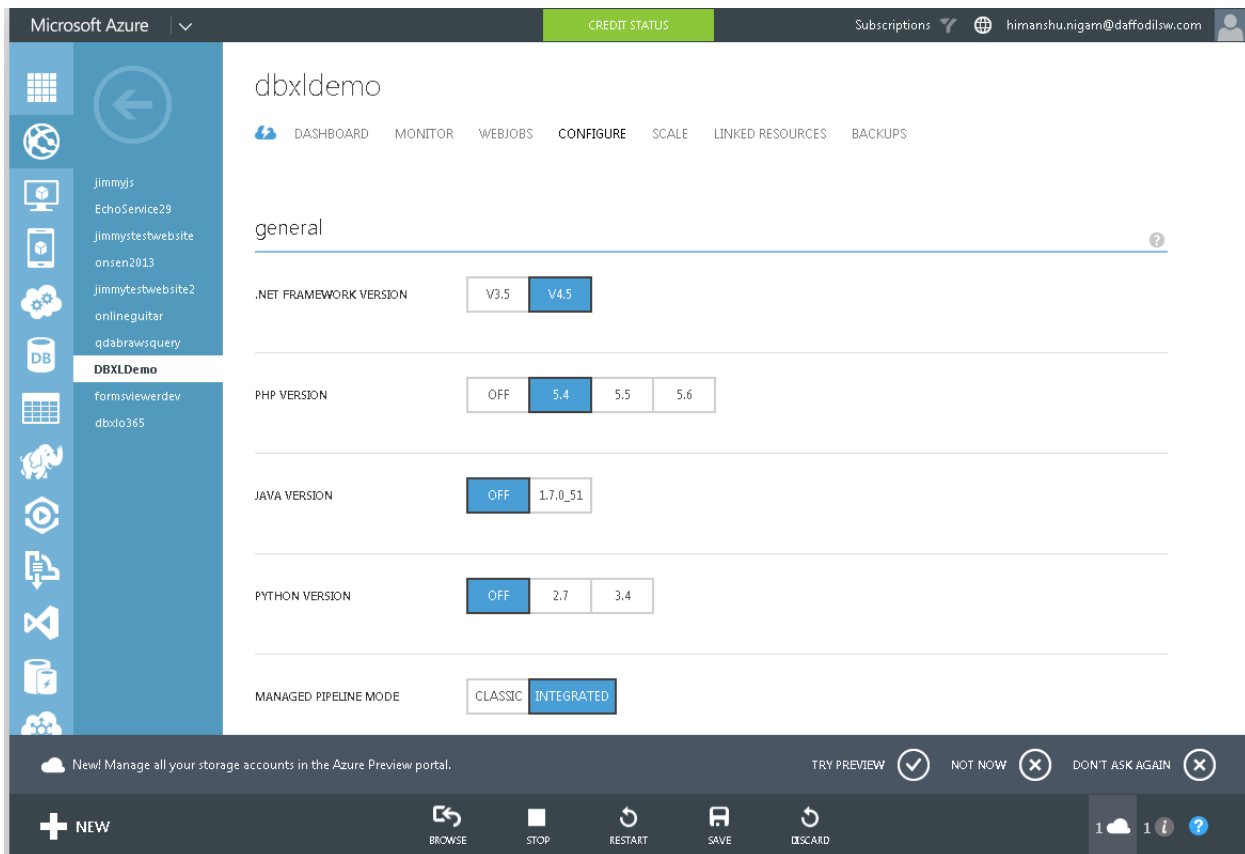


3. Note the following settings.

Azure web app configuration:

1. We have to add entries in azure web config
 - a. Client Id
 - b. Client secrets
 - c. Dbxl connection string for table storage.
 - d. ProgressNotificationEndpoint

To do this go to azure web app and click on configure.



2. Move down and find App Settings section and entries for:
 - a. ClientId = That we have generated in previous step "d3c31d7d-3f3c-4dda-89f7-bcecbd9bbc12"
 - b. ClientSecret = That we have generated in previous step "k0VO+x+3stjs99x/BWO/23DI6SUVAWeb2kG3sFcRPP4="
 - c. DbxlInstancesConnectionString = "DefaultEndpointsProtocol=https;AccountName=dbxlinstances;AccountKey=elg0/LPipSapxGBf4BGB7xIFLbTv7f7QsSDWgE1SQ1Xt5ykPuCHFrQ/h/6cNL33OnDNLa4WTAB3FO+8EtX70tg=="
 - d. ProgressNotificationEndpoint = basically this holds the path of Synchronization status page. Ex: <https://dbxleventreceiver.formsquo.com/Pages/SynchronizationStatus.aspx>.
3. Now add settings in connection strings section:
 - a. AzureWebJobsDashboard: Enter any storage connection string. We can use DbxlInstancesConnectionString for now. Choose custom setting in dropdown. Ex: DefaultEndpointsProtocol=https;AccountName=dbxlinstances;AccountKey=elg0/LPipSapxGBf4BGB7xIFLbTv7f7QsSDWgE1SQ1Xt5ykPuCHFrQ/h/6cNL33OnDNLa4WTAB3FO+8EtX70tg==
 - b. AzureWebJobsStorage: Enter any storage connection string. We can use DbxlInstancesConnectionString for now. Choose custom setting in dropdown. Ex: DefaultEndpointsProtocol=https;AccountName=dbxlinstances;AccountKey=elg0/LPipSap

xGBf4BGB7xIFLbTv7f7QsSDWgE1SQ1Xt5ykPuCHFrQ/h/6cNL33OnDNLa4WTAB3FO+8EtX70tg==

- c. DBXLConnectionString: This is the connection string of SQL Azure database connection string. Choose SQL database in dropdown.

Ex: Data Source=hrvlorfihz.database.windows.net;Initial

catalog=DbxlDocSyncProd;Uid=qdabraProduction@hrvlorfihz;Pwd=Qdabra@123

Create storage account connection string:

We need to create one storage account on Azure to save the Application data (DBXL instances and for saving error log info).

- i. To create Storage connection string click on New ->Data Services -> Storage -> Quick create.
- ii. Provide necessary parameter e.g. URL, Location affinity group, Replication.
- iii. Click on create storage account.

The screenshot shows the Microsoft Azure portal interface. On the left, the 'NEW' button is highlighted. In the center, the 'storage' section is active, showing a table with three storage accounts: 'dbxlinstances', 'jimmytestserver', and 'portalvhds18cz498ip1fhm'. All are 'Online'. On the right, the 'QUICK CREATE' form is open, showing fields for 'URL' (dbxlinstances), 'LOCATION/AFFINITY GROUP' (Central US), and 'REPLICATION' (Geo-Redundant). The 'CREATE STORAGE ACCOUNT' button is at the bottom right.

- iv. Now click on Storage -> Select your storage -> Click on Manage access key.

Microsoft Azure | CREDIT STATUS | Subscriptions | himanshu.nigam@daffodilsw.com

storage

NAME	STATUS	LOCATION	SUBSCRIPTION
dbxinstances	✓ Online	North Central US	Jimmy MSDN
jimmytestserver	✓ Online	East Asia	Jimmy MSDN
portalvhds18cz498ip1fhm	✓ Online	Japan West	Jimmy MSDN

New! Manage all your storage accounts in the Azure Preview portal.

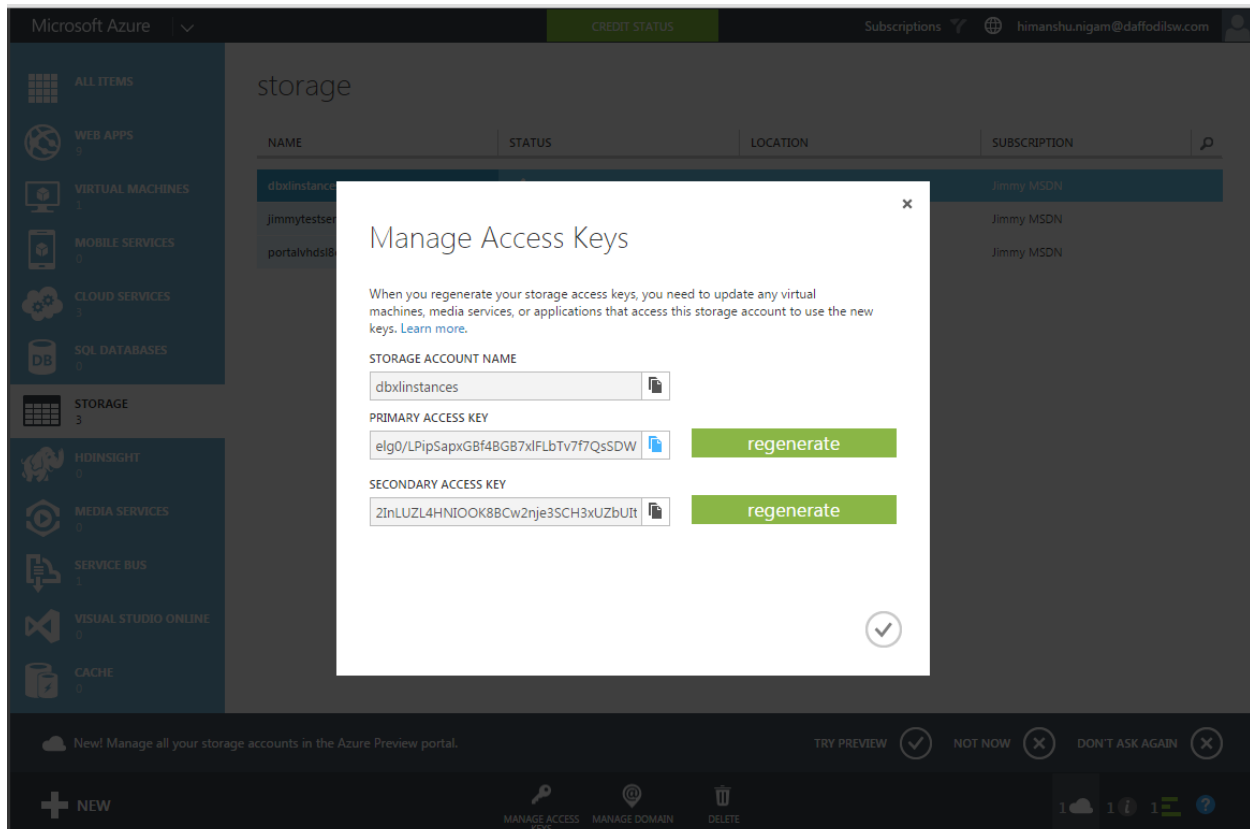
TRY PREVIEW ✓ NOT NOW ✗ DON'T ASK AGAIN ✗

+ NEW | MANAGE ACCESS KEYS | MANAGE DOMAIN | DELETE

v. Now copy your primary access key. And make connection string like below.

“DefaultEndpointsProtocol=https;AccountName=dbxinstances;AccountKey=elg0/LPipSapxGBf4BGB7xIFLbTv7f7QsSDWgE1SQ1Xt5ykPuCHFrQ/h/6cNL33OnDNLa4WTAB3FO+8EtX70tg==”

- vi. AccountName is storage url that we give in earlier step.
- vii. Account key is that we copied now.



viii. Give this string a name in web config.

The screenshot shows the Microsoft Azure portal interface. The left sidebar contains navigation icons and a list of resources including 'QdabraEventReceiv...', 'jimmytestwebsite2', 'onlineguitar', 'qdabrawsquery', 'DBXLDemo', 'formviewerdev', and the selected resource 'dbxlo365'. The main content area is titled 'monitoring' and includes a message: 'You can add endpoints only in Basic or Standard mode. [Learn more about configuring endpoint monitoring](#) or [Scale your site now](#).' Below this, the 'developer analytics' section shows 'PERFORMANCE MONITORING' set to 'OFF' with 'ADD-ON' and 'CUSTOM' options. The 'app settings' section displays a table of configuration values:

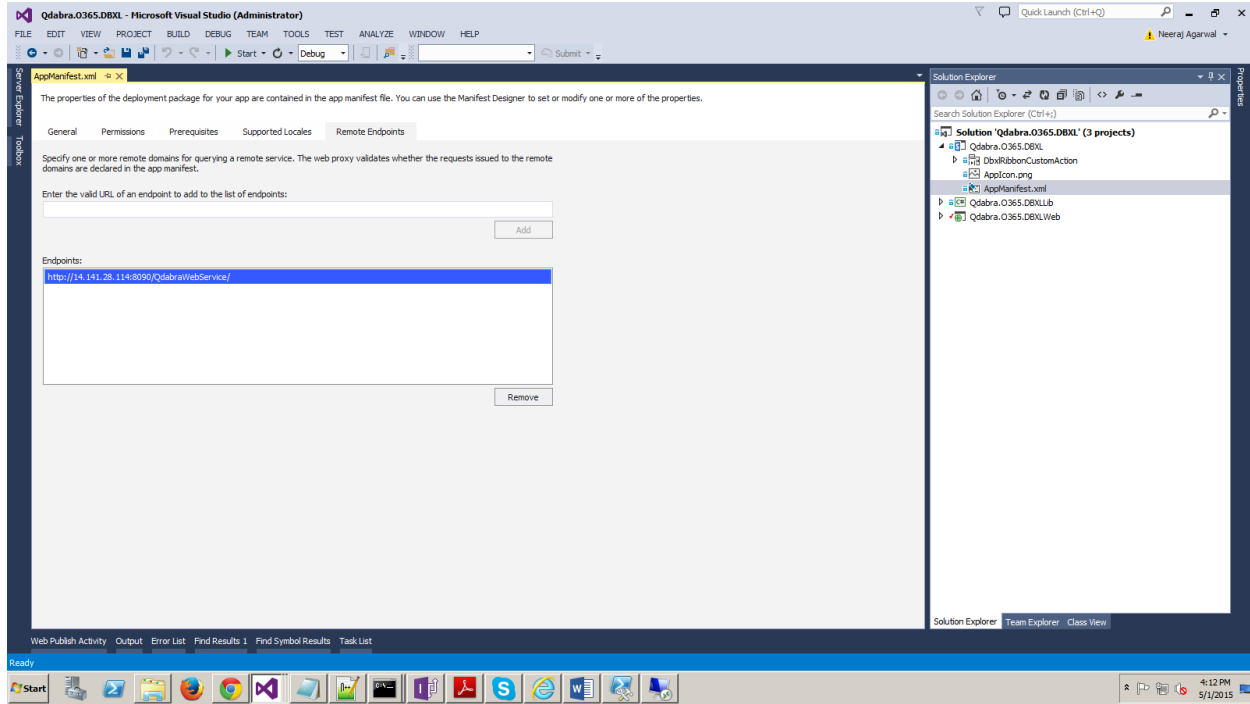
KEY	VALUE
WEBSITE_NODE_DEFAULT_VERSION	0.10.32
ClientId	174d9964-56f9-4ee8-93b2-ab18ebd2647f
ClientSecret	voIVRyYA6tqqdnVw9r7COa74zKWEoGT3cdk3zgEWgUU=
DbxInstancesConnectionString	DefaultEndpointsProtocol=https;AccountName=dbxinstances;Accou

Below the app settings is the 'connection strings' section, which includes a table with columns 'NAME' and 'VALUE', and a dropdown menu for 'SQL Database'. The bottom of the screen features a dark blue bar with a '+ NEW' button, 'BROWSE', 'STOP', and 'RESTART' buttons, and a notification area on the right with buttons for 'TRY PREVIEW', 'NOT NOW', and 'DON'T ASK AGAIN'.

Now web project is published and all the settings will be done for web project.

Create or Publish app package:

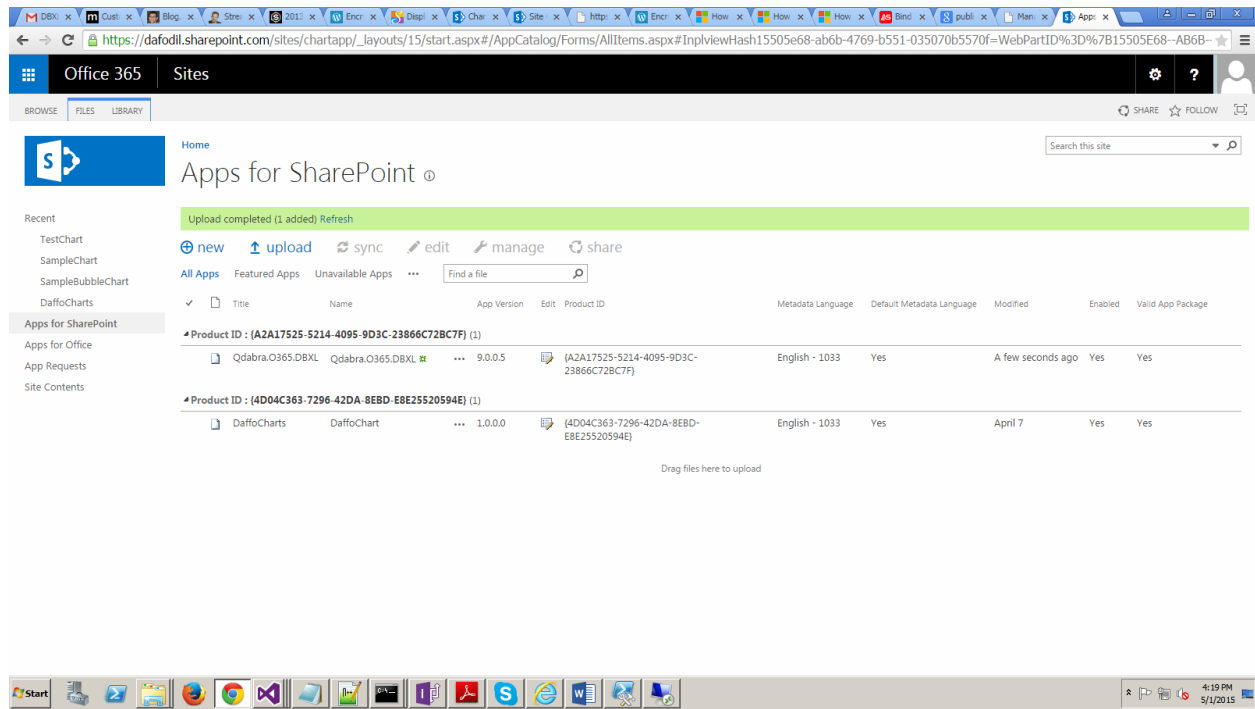
1. Now host the Qdabra DBXL web service and bind the service with a public IP.
2. Now copy the public IP Url and add it in the Remote Endpoints section of App manifest xml file.



3. Right click on app package and click on publish.
4. Then click Package the app and then enter the
 - a. Azure site Url.
 - b. Client Id and that we have generated earlier.
5. Click finish and copy the file path of app package.

Upload the app in App catalog on Office 365 account

1. Go to App catalog site (If it is not present then create a site collection using App Catalog template) and then upload the package here.



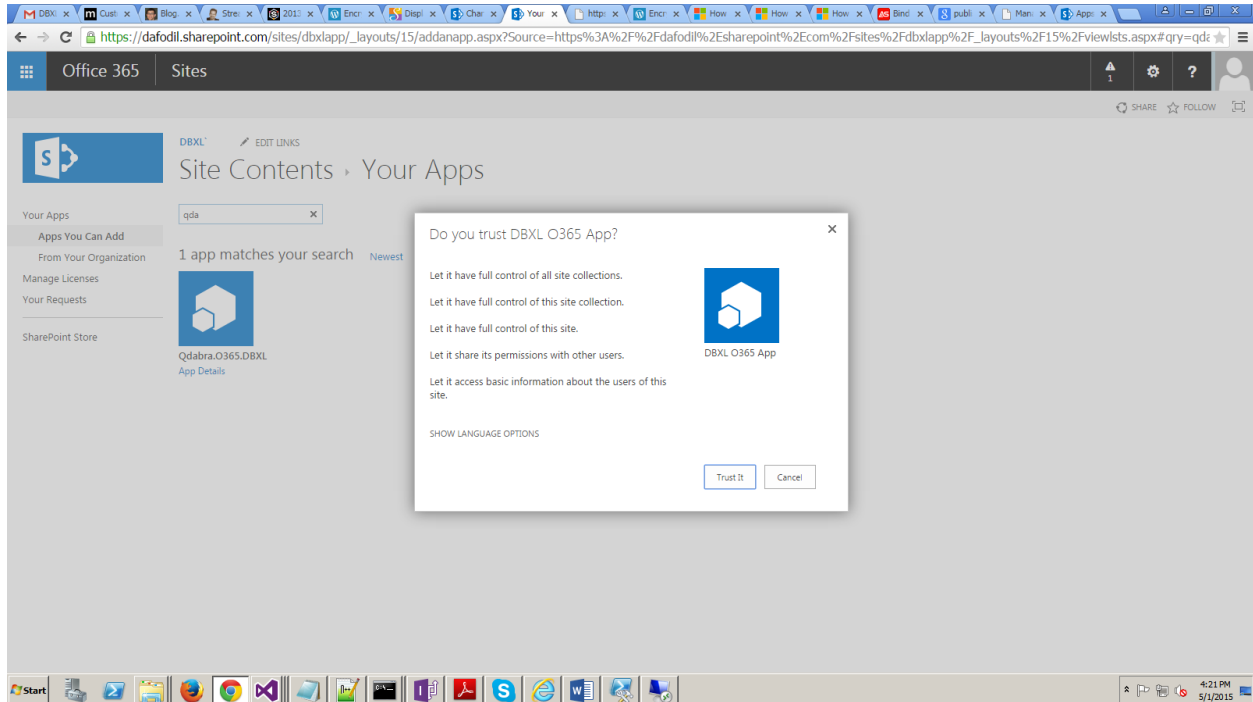
The screenshot shows the Office 365 App Catalog interface. A green banner at the top indicates "Upload completed (1 added) Refresh". Below this, the "All Apps" section displays a table of installed applications.

✓	Title	Name	App Version	Edit	Product ID	Metadata Language	Default Metadata Language	Modified	Enabled	Valid App Package
Product ID : {A2A17525-5214-4095-9D3C-23866C72BC7F} (1)										
	Qdabra.O365.DBXL	Qdabra.O365.DBXL	9.0.0.5		{A2A17525-5214-4095-9D3C-23866C72BC7F}	English - 1033	Yes	A few seconds ago	Yes	Yes
Product ID : {4D04C363-7296-42DA-8EBD-E8E25520594E} (1)										
	DaffoCharts	DaffoChart	1.0.0.0		{4D04C363-7296-42DA-8EBD-E8E25520594E}	English - 1033	Yes	April 7	Yes	Yes

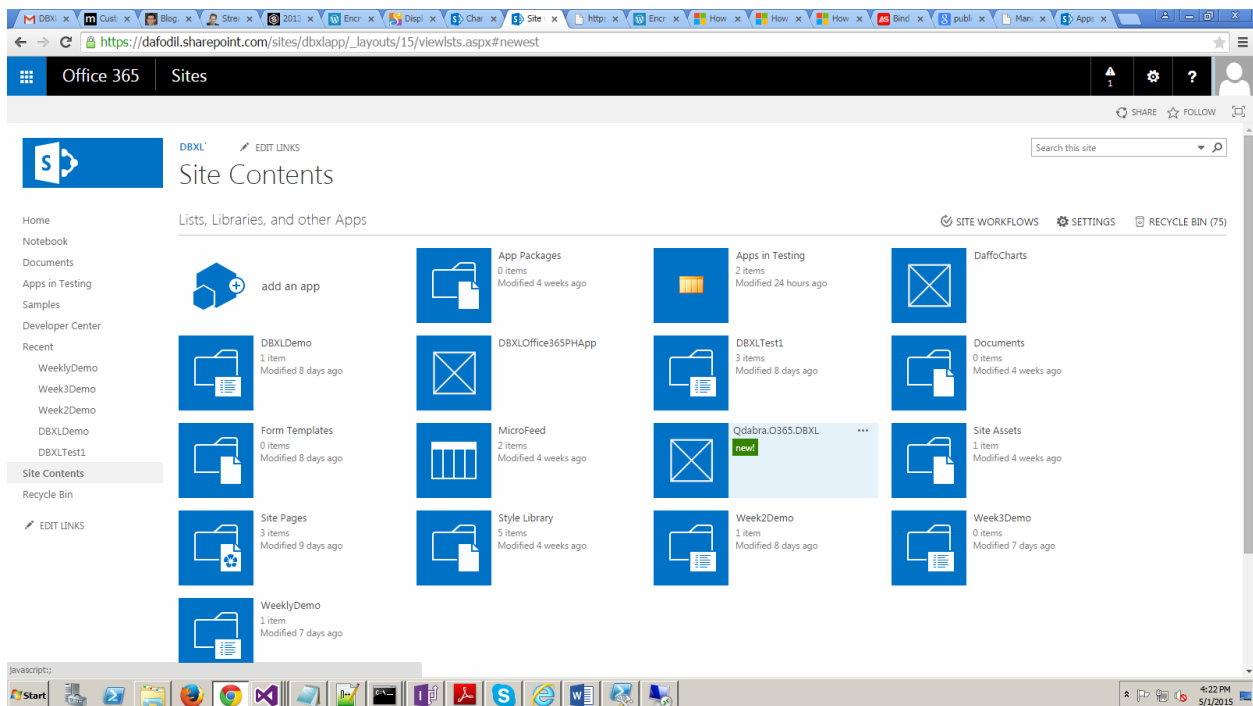
Drag files here to upload

Add the DBXL O365 app in the SharePoint site:

1. Now move to any site (<https://qdabra.sharepoint.com/demo>) click on Add an app -> find your app and Install it
2. Click on Trust It to provide access to the app.

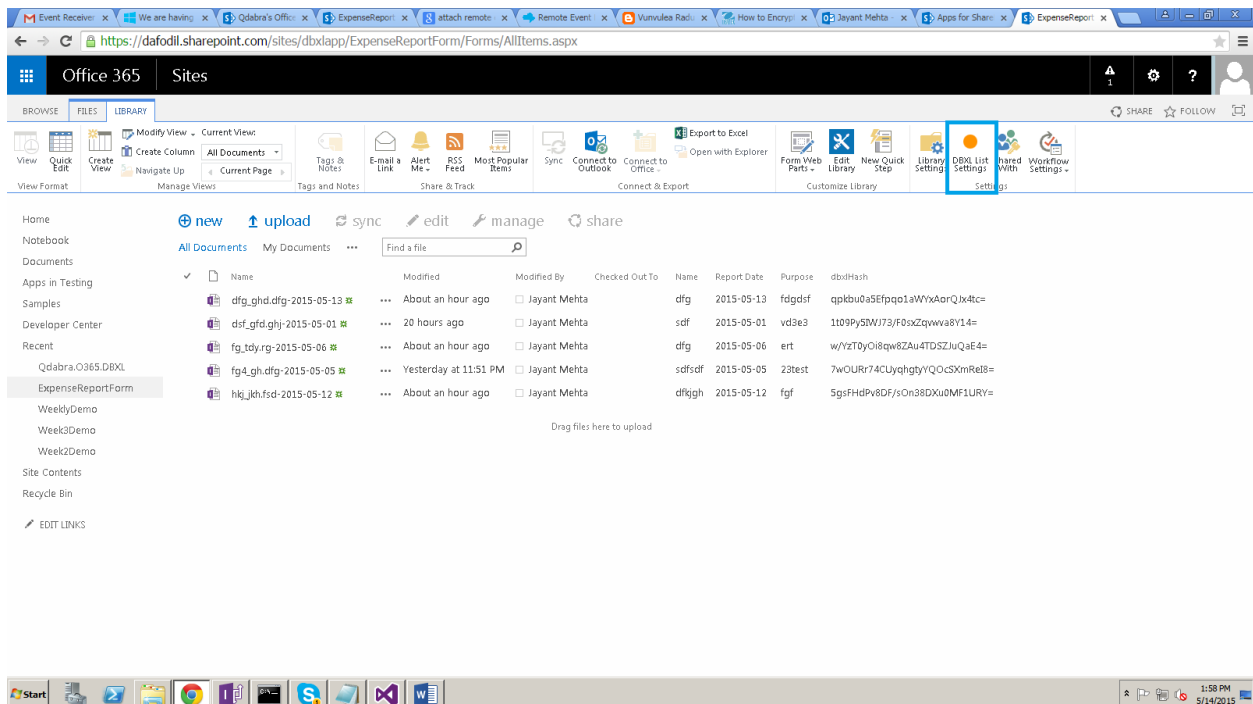


3. Now you can find your app in Site content of your site.



To Setup the Remote Event Receiver:

1. We will create a Form Library.
2. Create InfoPath form and configure the Submit data connection to point the Form Library
3. Publish the Form to SharePoint server.
4. Close the form.
5. Configure the Document Type in DBXL Configuration InfoPath Form.
6. Write a doctype name, upload the InfoPath form template, Do Database Connection setup.
7. Data table and columns mapping. (Same settings as On-Premise Dbxl Settings).
8. Let's get back to form library.
9. Click the DBXL List Settings from Ribbon button in form library settings.



10. Update the Remote End points of the DBXL web service in DBXL instance.

To submit document to DBXL, we need the DBXL service and that service Url should be in SharePoint app package to manage trust of DBXL service with SharePoint. So if we want to add/Edit DBXL service Url then we need to create another app package with the added/updated service Url (Remote End Points). To automate this process we have added a page that will take the Remote end points and will create the new app package with the added/updated remote end points.

Follow the below mentioned steps to update Remote end points:

- a. To update remote end points click on DBXL instance link on DBXL List Settings page.

DBXL Settings: ExpenseReportForm

DBXL Service: Enter the configuration details.

DBXL Instance: Dev

[Manage DBXL Instances](#)

Document Settings: Enter the xpaths containing the document properties. These are passed to DBXL on save or update.

Document Type Name: WeeklyDemo

Document Type Name Xpath: Enter the Document Type Name Xpath

Conditional Submit: Example: /my/myFields/my.Value = '1'

Note: the condition is not validated.

Conditional Delete: Example: /my/myFields/my.Value = '1'

Note: the condition is not validated.

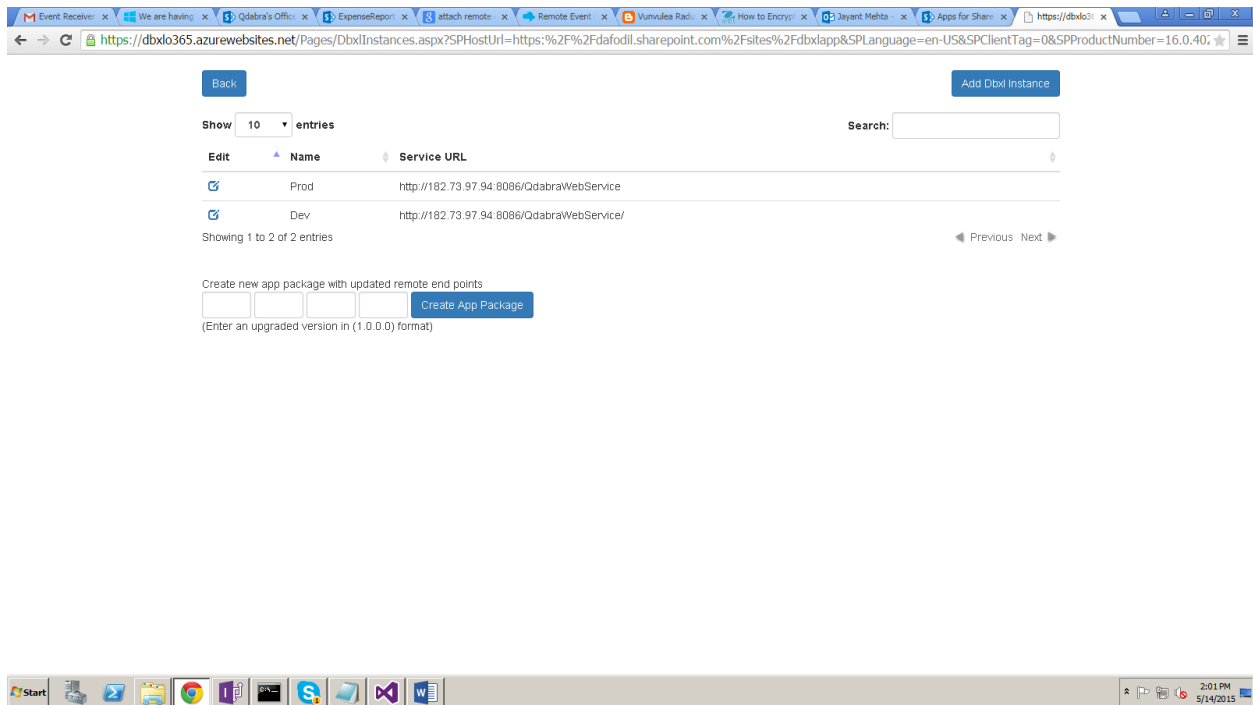
Name XPath: /my/ExpenseReport/my.Main/my.ExpenseDetails/my.Employee/my.Name

Note: xpaths are not validated.

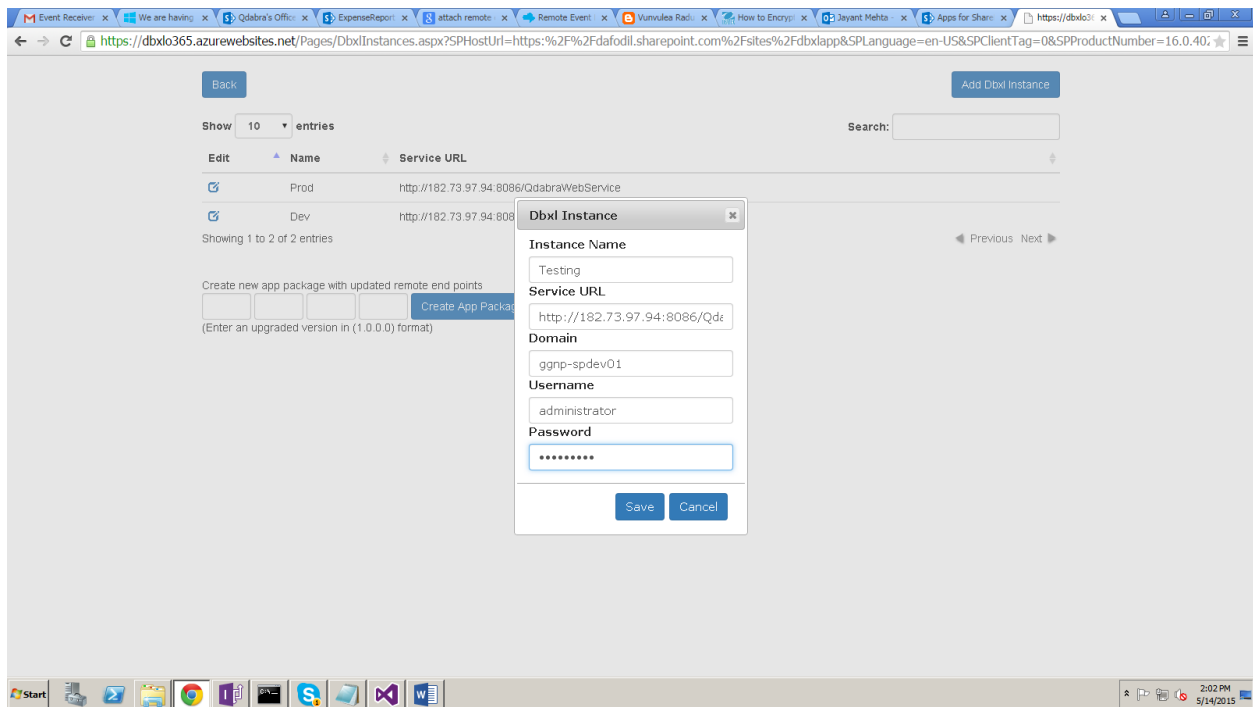
Author XPath: Enter the Author Xpath

- b. Now dbxl instance page will open.

DBXL installation/Configuration and upgrade



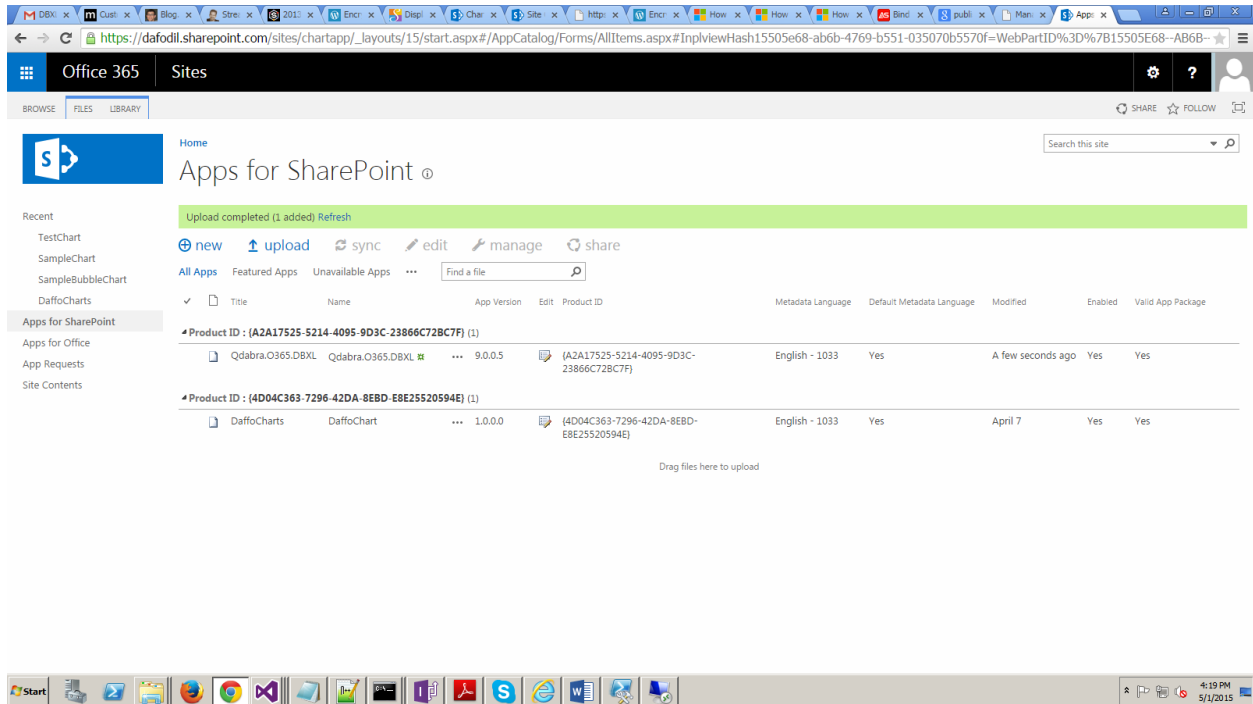
- c. Now click on Add/Edit Dbxl Instance button to add new dbxl instance provide the all parameters.



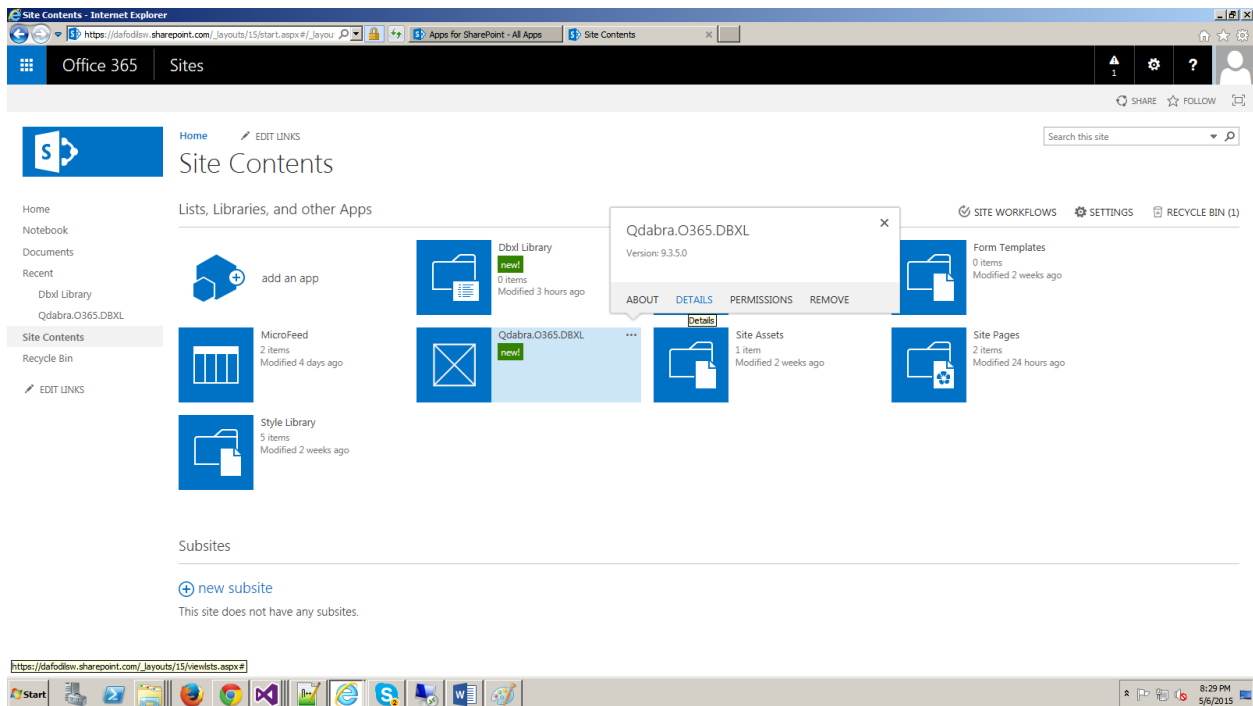
- d. On Save entries will show in data table.

DBXL installation/Configuration and upgrade

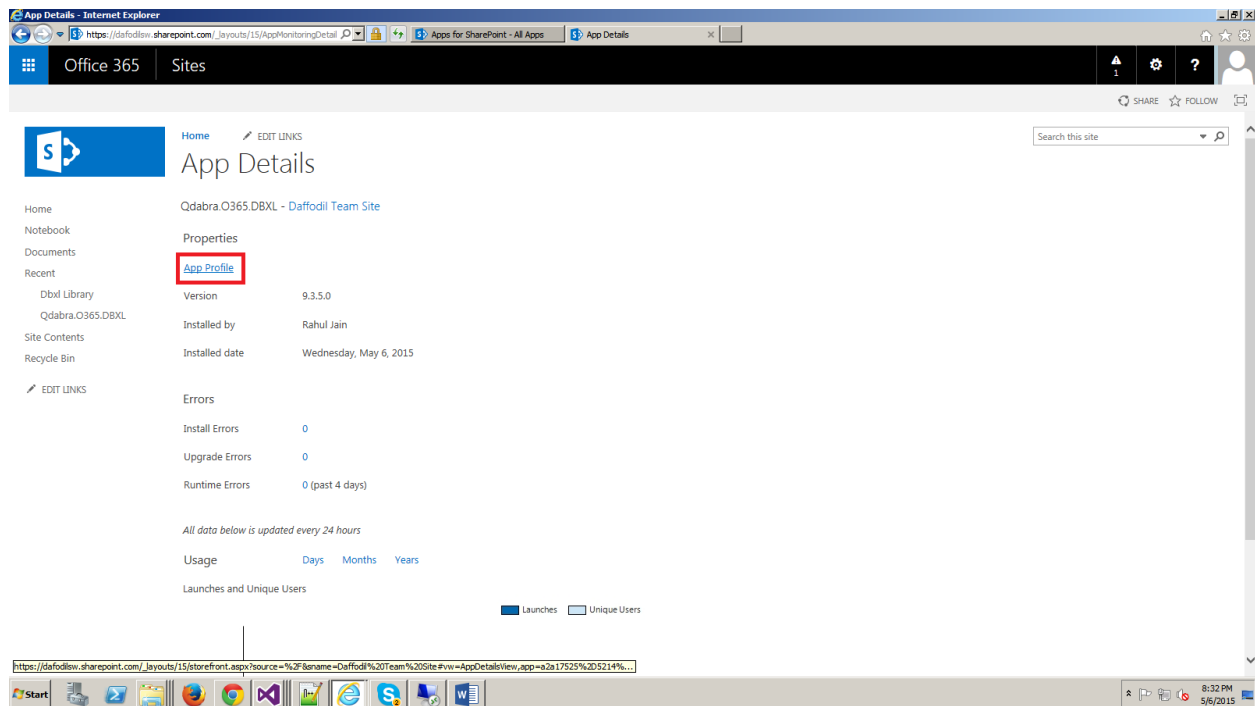
- e. Now to create the package provide the app version and click on Create App Package.
- f. Then it will download the new app package.
- g. Now go to app catalog and upload the new app package.



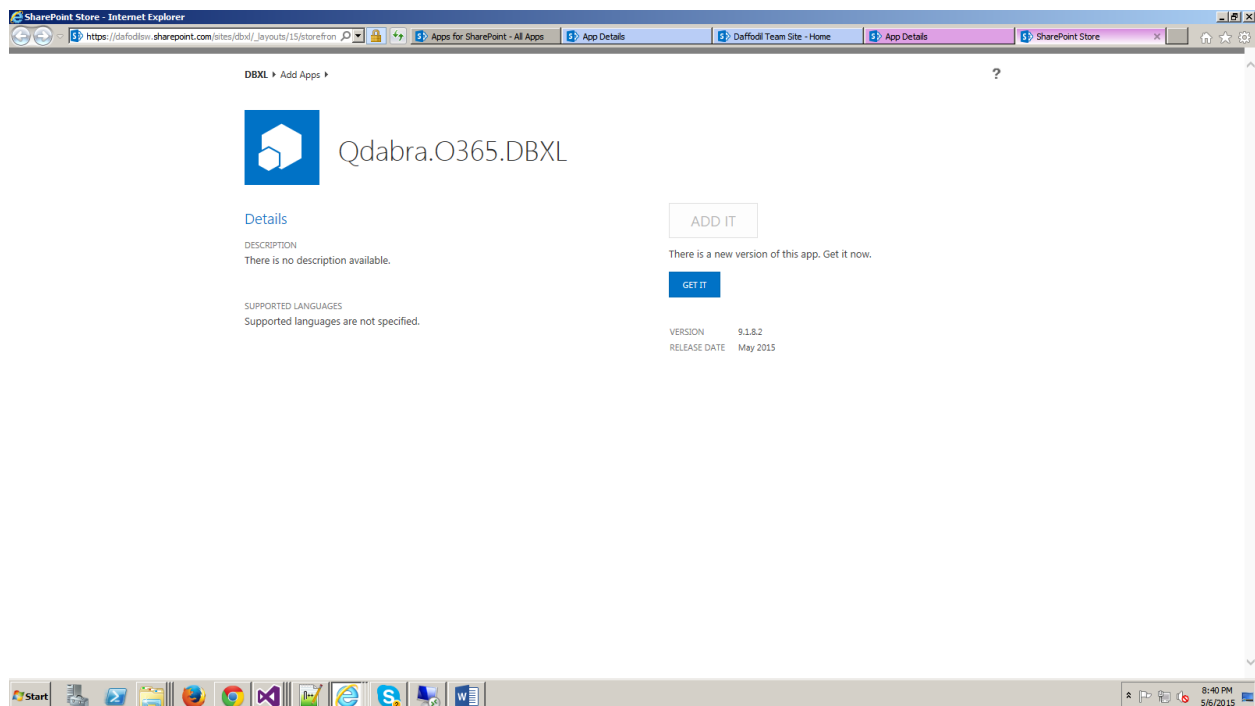
- h. Now to update the package to particular SharePoint site.
- i. Go to site content -> your app and click on app details.



j. Now click on app profile link.



k. Now Click on Get it to update the app.



11. Do the necessary settings required for DBXL settings.

DBXL Settings: **WeeklyDemo**

DBXL Service: DBXL Instance:

Enter the configuration details.

Test

[DBXL Instance](#)

Document Settings: **Document Type Name:**

Enter the xpaths containing the document properties. These are passed to DBXL on save or update.

WeeklyDemo

Document Type Name XPath

Enter the Document Type Name XPath

Conditional Submit:

Example: /my:myFields/my:Value = '1'

Note: the condition is not validated.

Conditional Delete:

/my:expenseReport/my:items/my:total > 1

Note: the condition is not validated.

Name XPath:

/my:expenseReport/my:employee/my:name

Note: xpaths are not validated.

Author XPath:

/my:expenseReport/my:employee/my:name

Description XPath:

/my:expenseReport/my:items/my:item/my:description

Notifications :

Specify whom to notify when errors occur submitting forms to DBXL.

jayant@dafodil.onmicrosoft.com

Enable DBXL Submit: ☒ Enable DBXL Submit

Enable or disable submitting to DBXL on

Below are the complete description of each DBXL List setting.

- DBXL Instances: These are the services Url for which we have configured our DBXL to submit the documents.
- Document type name: This is the doc type name that we have configured in DBXL administration form.
- Document type name XPath: Xpath of doc type.
- Conditional submit: Condition entered in this will check on document submit and if the condition pass then document will submit to DBXL if not then it check whether the document is already submitted if it found then delete the document from DBXL.
- Conditional delete: Condition entered in this will check on document submit and if the condition pass then document will be deleted from DBXL and SharePoint both if not then document is deleted from SharePoint only.

- f. Name XPath: XPath of the document field that we want to show in Name field of DBXL administration form.
- g. Author XPath: XPath of the document field that we want to show in Author field of DBXL documents.
- h. Description XPath: XPath of the document field that we want to show in the Description field of DBXL documents.
- i. Notifications: This takes comma delimited email address for sending notifications for synchronization and errors.
- j. Enable DBXL submit: This check box will handle DBXL document submit. If unchecked then no document will be submit to DBXL.
- k. Force DBXL submit: If user always wants to submit document whether is changed or not, then this checkbox will override the functionality of preventing document submit if it is not changed.

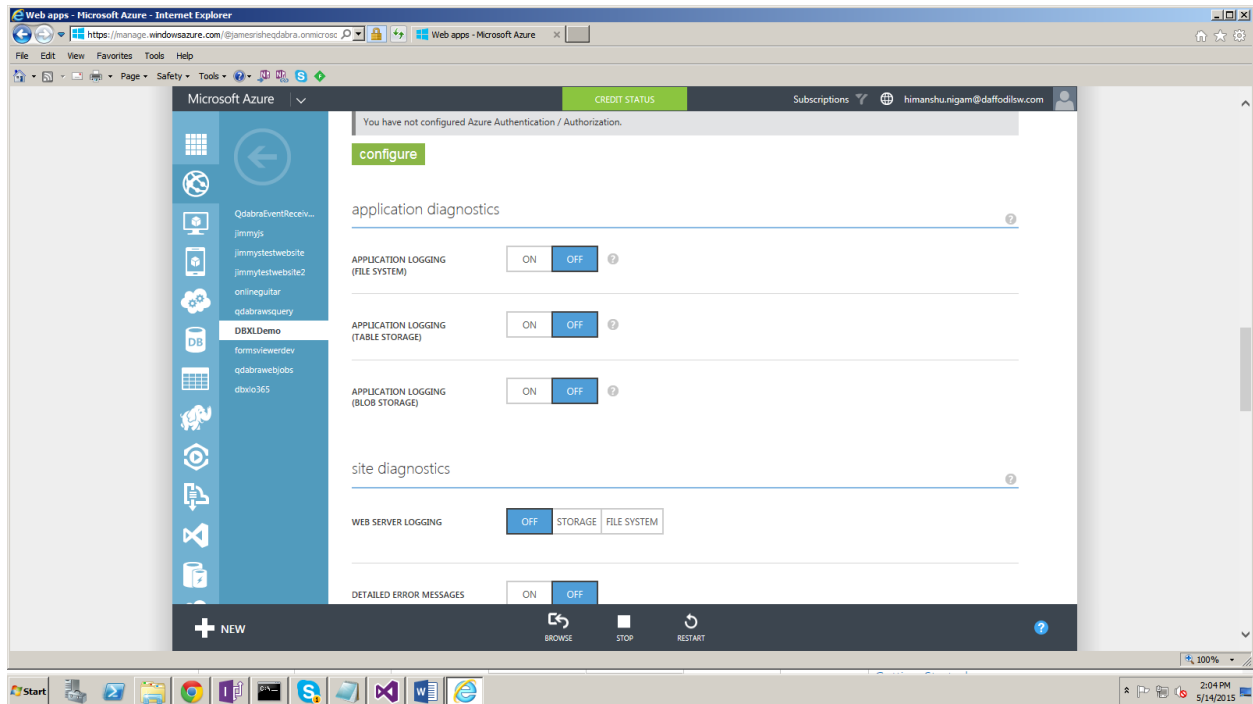
List settings page buttons

- a. Synchronize: This will synchronize the all library documents regardless of whether “Enable DBXL submit” is checked or not but this will always check for other list settings.
- b. Save and Test: This will save the DBXL settings to library properties bag and validate for the DBXL list settings and service.
- c. OK: This button save the DBXL list settings to library properties bag and redirect to library settings page.
- d. Cancel: This button cancel the current operation and redirect to library settings page.

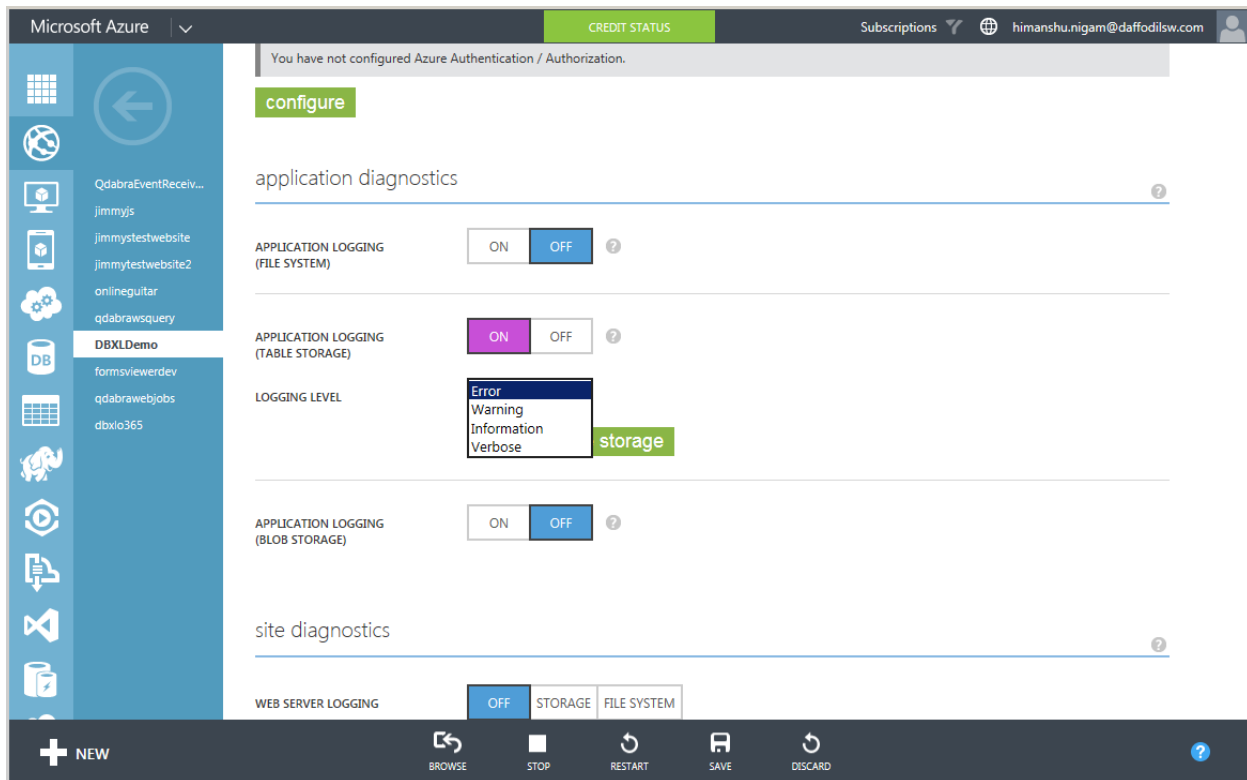
To log application error:

To enable or track the application specific error you need to do the following settings on Azure.

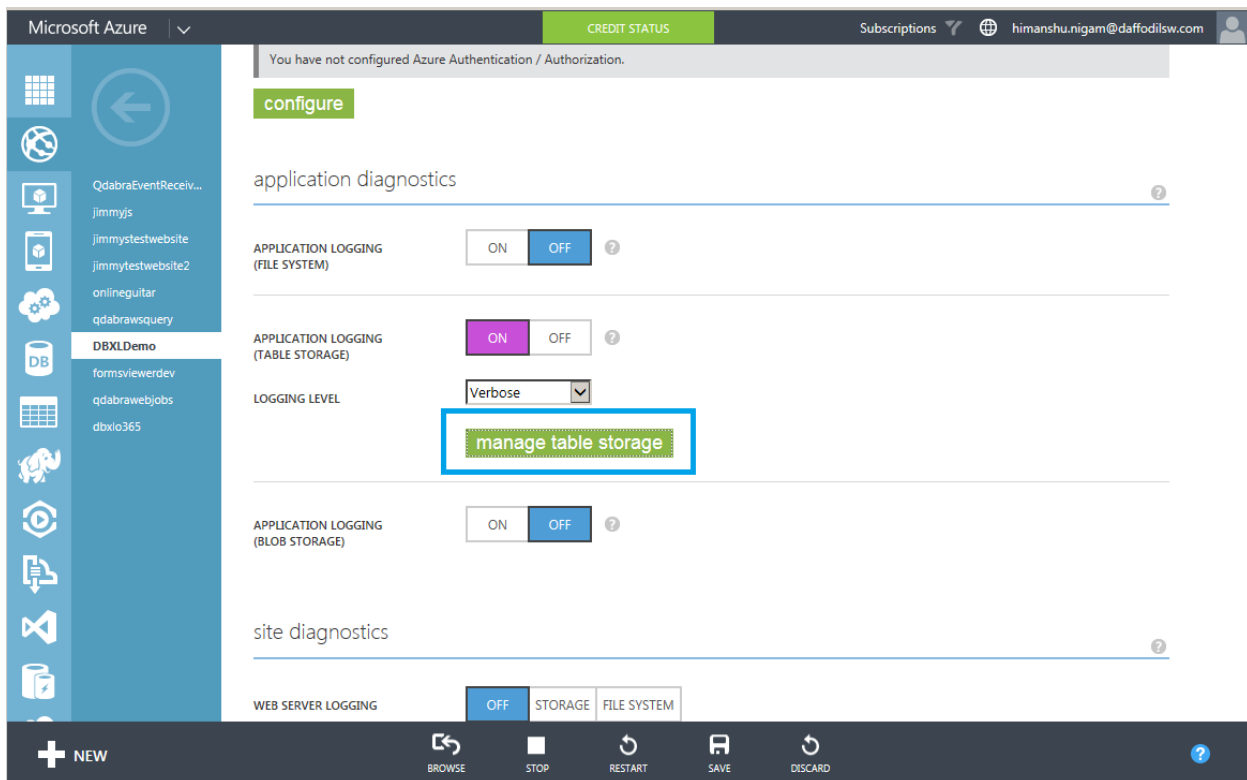
1. Login to Azure, Move to web application and click on configure.
2. Scroll down and find application diagnostics section.



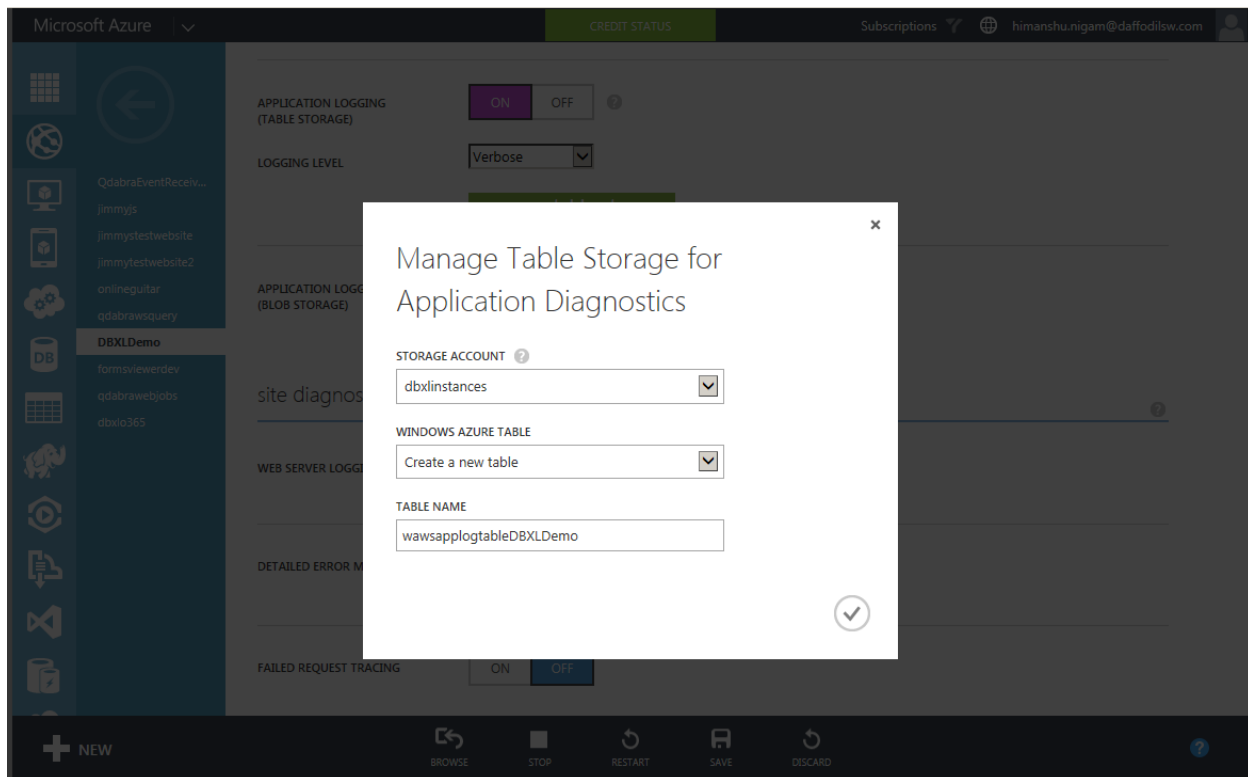
3. Now turn ON Application logging (Table storage).
4. Now select logging level. In our application we have set it to verbose to read all application errors.
5. If you select information then only errors, warning and information will be logged.



6. Now click on Manage table storage to storage logging information.



7. Now provide
 - a. A storage account name. You can use the same as we created earlier.
 - b. Window azure table you can either create or can use any other table.
 - c. If you select create a new table then provide a name for the table the click on Check arrow to create your table.

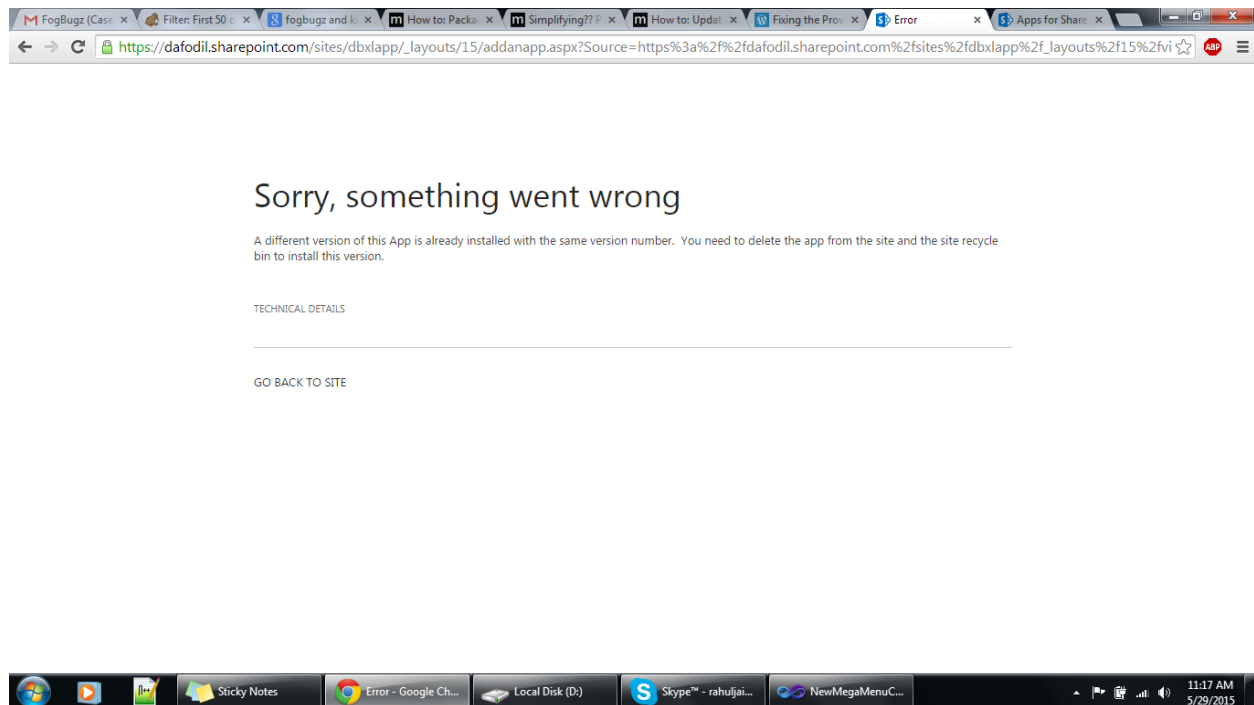


8. Now application logs can be viewed in any storage explorer.

Troubleshooting:

1. Installing App with same version number:

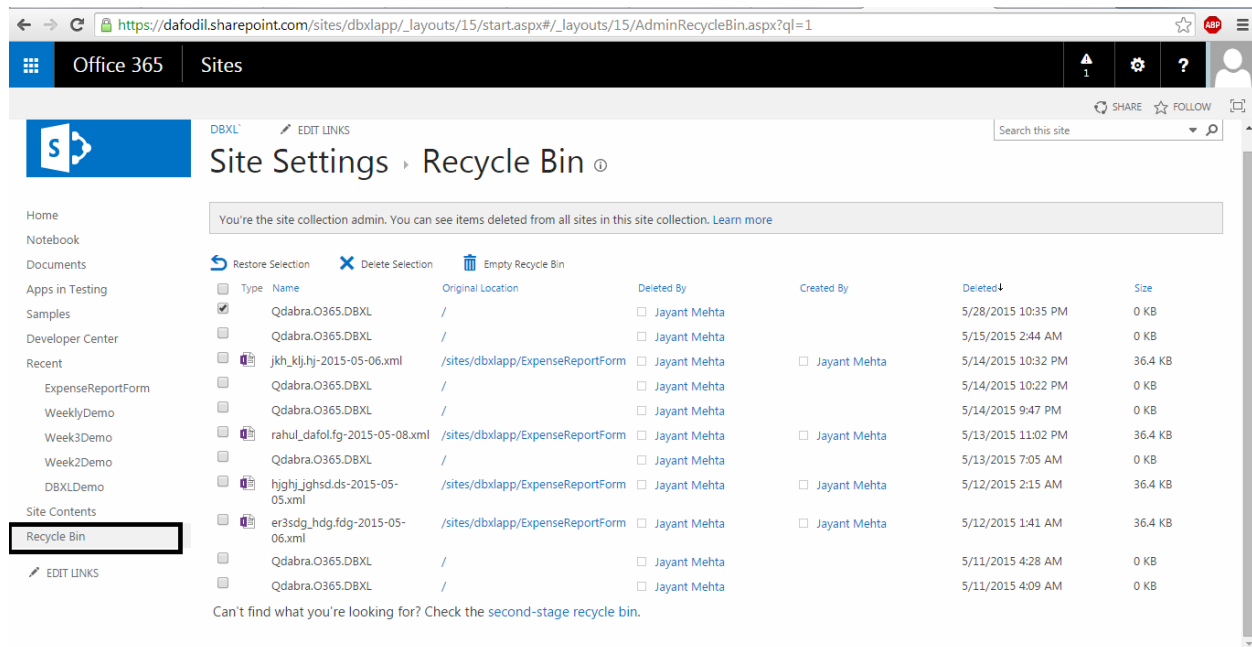
Suppose you have updated the remote end points and created a new app package with same number. But when you are trying to update the app package you got the following error.



Error is “A different version of this App is already installed with the same version number. You need to delete the app from the site and the site recycle bin to install this version.”

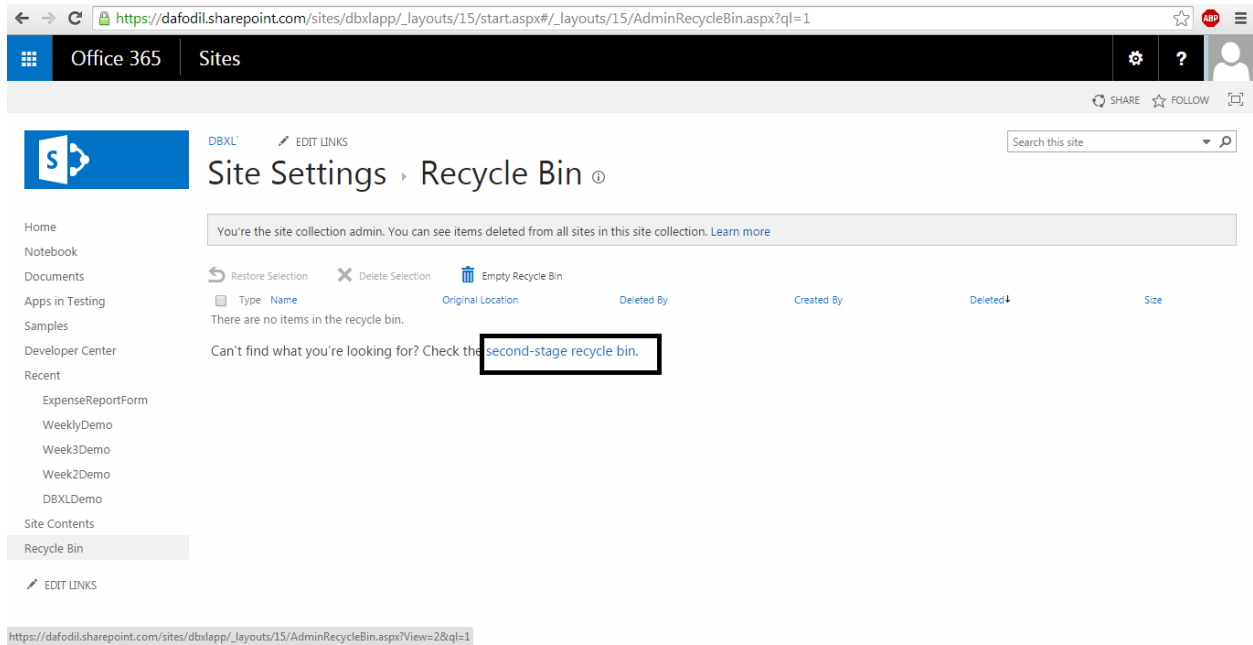
Error briefly explains the problem that we have already an app is installed with the same version number actually SharePoint is not removing the app from the site, it just puts it in the site’s recycle bin. So the app really does live on, and that’s why you can’t add it again. So the solution is either we have to update the app with new version number that is greater than the previous or to install the app with same version number we have to remove the app from the both Primary and Secondary recycle bin. Below are the steps explaining how to achieve the same.

1. Open SharePoint site on which App is installed previously and click on Recycle Bin from the left.



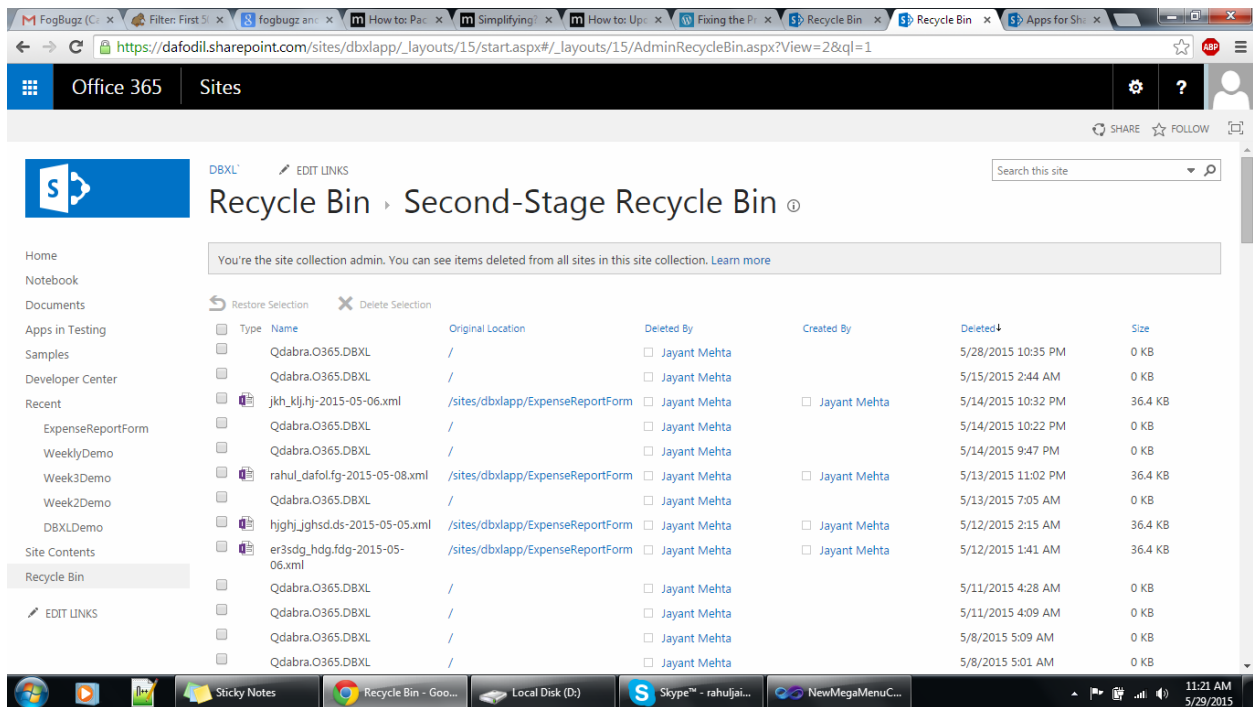
2. Remove your app package from here by selecting the particular app package and click on Delete Selection button.
3. Only deletion for this doesn't work you will still get the same error if trying to install the app. Because SharePoint still save it in Secondary recycle bin. Now to completely remove the app from SharePoint click on "second stage recycle bin" from the same page.

DBXL installation/Configuration and upgrade



The screenshot shows the SharePoint 'Recycle Bin' settings page. The breadcrumb navigation is 'Site Settings > Recycle Bin'. A message states: 'You're the site collection admin. You can see items deleted from all sites in this site collection. [Learn more](#)'. Below this, there are buttons for 'Restore Selection', 'Delete Selection', and 'Empty Recycle Bin'. A table header is visible with columns: 'Type', 'Name', 'Original Location', 'Deleted By', 'Created By', 'Deleted', and 'Size'. A text prompt says 'There are no items in the recycle bin.' and 'Can't find what you're looking for? Check the [second-stage recycle bin.](#)'. The left sidebar shows 'Recycle Bin' as the selected option under 'Site Contents'. The URL at the bottom is 'https://dafodil.sharepoint.com/sites/dbxlapp/_layouts/15/AdminRecycleBin.aspx?View=2&ql=1'.

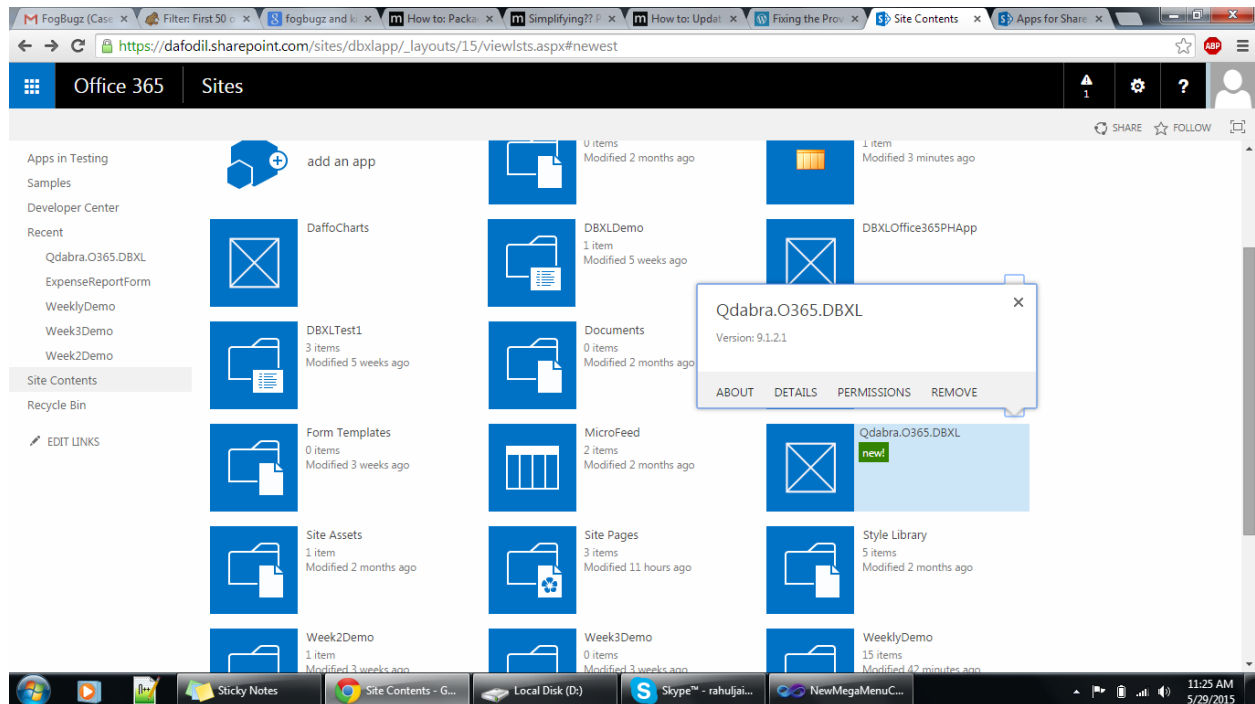
4. Now find your app and remove it.



The screenshot shows the 'Second-Stage Recycle Bin' in SharePoint. The breadcrumb navigation is 'Recycle Bin > Second-Stage Recycle Bin'. A message states: 'You're the site collection admin. You can see items deleted from all sites in this site collection. [Learn more](#)'. Below this, there are buttons for 'Restore Selection' and 'Delete Selection'. A table lists deleted items with columns: 'Type', 'Name', 'Original Location', 'Deleted By', 'Created By', 'Deleted', and 'Size'. The table contains several entries, including 'Qdabra.O365.DBXL' and XML files. The left sidebar shows 'Recycle Bin' as the selected option. The URL at the bottom is 'https://dafodil.sharepoint.com/sites/dbxlapp/_layouts/15/AdminRecycleBin.aspx?View=2&ql=1'.

Type	Name	Original Location	Deleted By	Created By	Deleted	Size
	Qdabra.O365.DBXL	/	Jayant Mehta		5/28/2015 10:35 PM	0 KB
	Qdabra.O365.DBXL	/	Jayant Mehta		5/15/2015 2:44 AM	0 KB
	jkh_kjh-hj-2015-05-06.xml	/sites/dbxlapp/ExpenseReportForm	Jayant Mehta	Jayant Mehta	5/14/2015 10:32 PM	36.4 KB
	Qdabra.O365.DBXL	/	Jayant Mehta		5/14/2015 10:22 PM	0 KB
	Qdabra.O365.DBXL	/	Jayant Mehta		5/14/2015 9:47 PM	0 KB
	rahu_dafolfg-2015-05-08.xml	/sites/dbxlapp/ExpenseReportForm	Jayant Mehta	Jayant Mehta	5/13/2015 11:02 PM	36.4 KB
	Qdabra.O365.DBXL	/	Jayant Mehta		5/13/2015 7:05 AM	0 KB
	highj_ghsd.ds-2015-05-05.xml	/sites/dbxlapp/ExpenseReportForm	Jayant Mehta	Jayant Mehta	5/12/2015 2:15 AM	36.4 KB
	er3sdg_hdg.fdg-2015-05-06.xml	/sites/dbxlapp/ExpenseReportForm	Jayant Mehta	Jayant Mehta	5/12/2015 1:41 AM	36.4 KB
	Qdabra.O365.DBXL	/	Jayant Mehta		5/11/2015 4:28 AM	0 KB
	Qdabra.O365.DBXL	/	Jayant Mehta		5/11/2015 4:09 AM	0 KB
	Qdabra.O365.DBXL	/	Jayant Mehta		5/8/2015 5:09 AM	0 KB
	Qdabra.O365.DBXL	/	Jayant Mehta		5/8/2015 5:01 AM	0 KB

5. After deleting the app from secondary recycle bin try to add your app from site content then your will install with the same app version.



Appendix: SyncStatus.sql

```
USE [dbxldocsyncdev]
GO

/***** Object: Table [dbo].[SyncStatus]      Script Date: 6/7/2016
17:58:23 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[SyncStatus] (
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [ListId] [nvarchar](255) NOT NULL,
    [UserId] [nvarchar](255) NULL,
    [StartDate] [datetime] NULL,
    [LastModifiedDate] [datetime] NULL,
    [EndDate] [datetime] NULL,
    [Status] [nvarchar](512) NULL,
    [JobId] [nvarchar](255) NULL,
    [DocumentsProcessed] [int] NULL,
    [TotalDocuments] [int] NULL,
```

```
        [ErrorCount] [int] NULL,  
PRIMARY KEY CLUSTERED  
(  
        [ID] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =  
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)  
)  
  
GO
```