# OfFICE PNP: Prevent Site Deletion

|  |  |
| --- | --- |
| Summary: | Applies to: |
| This sample shows how to create a sandbox solution for preventing site deletion and using an app to deploy the solution with an application | * Office 365 Multi-Tenant (MT) |
| Solution: | Provisioning.PreventDeleteSites, version 1.0 |
| Author: | Suman Chakrabarti & Frank Marasco (Microsoft) |
| //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // THIS CODE IS PROVIDED \*AS IS\* WITHOUT WARRANTY OF  // ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING ANY  // IMPLIED WARRANTIES OF FITNESS FOR A PARTICULAR  // PURPOSE, MERCHANTABILITY, OR NON-INFRINGEMENT.  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | |

# Introduction

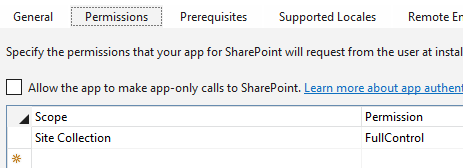
This solution combines a solution which is still partial-trust code (by using a sandbox solution) with a new provider-hosted application. One of the challenges of current remote event receivers is that they do not support a synchronous event processing. This means that when the SiteDeleting, WebDeleting, or ListDeleting events would not be preventable the event would not be able to cancel the site from being deleted before it is deleted.

This samples demonstrates:

* Development of a simple sandboxed solution using an SPSiteEventReceiver to prevent site deletion.
* Deployment of a solution to the Solution Gallery.
* Activation of a solution in the Solution gallery.
* Deactivation of a solution in the Solution gallery.
* Removal of a solution from the Solution gallery.

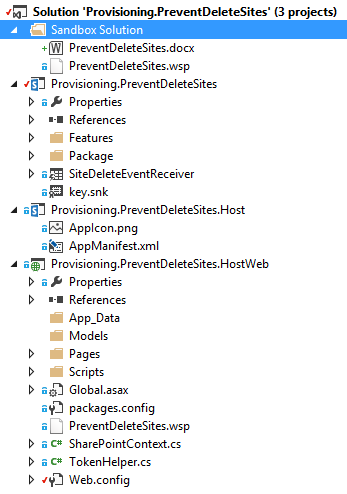
# Security

Permissions for this solution require that the solution have Site Collection Full Control to be able to deploy and activate the sandboxed solution.



# Solution

The process outlined in the sample describes Deployment, Activation, Testing deletion, Deactivation, and Removal of the solution.



## Sandbox Solution

The sandbox solution is pretty simple. It consists of an event receiver wired up for **SiteDeleting** and **WebDeleting**. When the user (or system process) attempts to delete a site collection or web application, the event receiver cancels that action.

public override void SiteDeleting(SPWebEventProperties properties) {

properties.Cancel = true;

properties.ErrorMessage = "Site collection cannot be deleted";

}

public override void WebDeleting(SPWebEventProperties properties) {

properties.Cancel = true;

properties.ErrorMessage = "Site cannot be deleted";

}

## Solution Deployment/Removal

The solution deployment consists of uploading the wsp to the Solution Gallery. This uses the FileCreationInformation object to upload the file to the gallery. Removing the site is done by finding the uploaded file and calling the DeleteObject method.

// get the file from the server path in the provider site

var filePath = Server.MapPath("~/PreventDeleteSites.wsp");

var file = new FileStream(filePath, FileMode.Open);

// create the FileCreationInformation object and prepare

// to upload it to the solution gallery

var fileCI = new FileCreationInformation() {

ContentStream = file,

Url = "PreventDeleteSites.wsp",

Overwrite = true

};

// upload the solution to the gallery

var uploadedFile = solutionGallery.RootFolder.Files.Add(fileCI);

clientContext.Load(uploadedFile);

clientContext.ExecuteQuery();

## Solution Activation/Deactivation

Solution activation is done using the DesignPackage objects to determine the wsp solution package. DesignPackage.Install and DesignPackage.Uninstall are used to activate and deactivate the solution for the site collection, respectively.

// get the DesignPackageInfo (which is the same name for a sandbox solution)

var wsp = new DesignPackageInfo(){

// during deployment, the solution ID is not necessary

PackageGuid = Guid.Empty, // 4c16c0b9-0162-43ad-a8e9-a4b810e58a56

PackageName = "PreventDeleteSites"

};

// install the solution from the file url

var filerelativeurl = solutionGallery.RootFolder.ServerRelativeUrl + "/PreventDeleteSites.wsp";

DesignPackage.Install(clientContext, clientContext.Site, wsp, filerelativeurl);

clientContext.ExecuteQuery();

# Dependencies

* Microsoft.SharePoint.Client.dll
* Microsoft.SharePoint.Client.Runtime.dll
* Microsoft.SharePoint.Client.Publishing.dll
* [Setting up provider hosted app to Windows Azure for Office365 tenant](http://blogs.msdn.com/b/vesku/archive/2013/11/25/setting-up-provider-hosted-app-to-windows-azure-for-office365-tenant.aspx)