

Import a Workflow

Last Modified on 04/24/2017 3:50 am EDT

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Xml.Linq;
using PNMsoft.Sequence.Deployment;
using PNMsoft.Sequence.Diagnostics;
using PNMsoft.Sequence.Runtime;
using PNMsoft.Sequence.Security;
using PNMsoft.Sequence.Utility;

namespace API.ImportWorkflow
{
    class Program
    {
        static void Main(string[] args)
        {
            WorkflowRuntime runtime = null;
            runtime = new WorkflowRuntime();
            runtime.Run(typeof(WorkflowEngine));
            IWorkflowEngine engine = WorkflowRuntime.Engine;
            AuthenticatedUser user = engine.GetServiceWithCheck<IAuthenticationService>().Authenticate("adminstrator", "e");
            //AuthenticatedUser user = WorkflowRuntime.Engine.GetService().Authenticate();
            SecurityManager.Impersonate(user);
            using (SecurityManager.CreateContext(user))
            {
                IDeploymentService ds = engine.GetService<IDeploymentService>();
                string packName = Directory.GetCurrentDirectory().ToString() + @"\Workflows/Basic Form Design Sample-0.5.zip";
                ImportWF(ds, packName);
            }

            internal static void ImportWF(IDeploymentService depService, stringpackageName)
            {
                string sharedResourcesPath =
                System.Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles) +
                @"\PNMsoft\Shared Resources";
                try
                {
                    string[] packages = GetFiles(@"Workflows");
                    if (packages.Length != 0)
                    {
                        //LoadEngine();
                        IDeploymentService deployService =WorkflowRuntime.Engine.GetServiceWithCheck<IDeploymentService>
                    );

                    DiagnosticUtility.Designtime.TraceInformationEvent("Starting import Wfs");

                    foreach (string package in packages)
                    {
                        string xml = depService.GetObjectsListFromPackage(packageName, ImportType.Import);
                        XDocument doc = XDocument.Parse(xml);
                        XElement workflowElements = doc.Root.Elements("Section").FirstOrDefault(el => el.Attribute("name")
                    ).Value == "WorkflowElements");
                        Guid workflowId = Guid.Empty;
```

```

if (workflowElements != null)
{
if (!GuidUtil.TryParse(workflowElements.Element("Workflow").Attribute("id").Value, out workflowId)
)
{
throw new Exception("Invalid package.");
}
}

IWorkflowDefinitionService service = WorkflowRuntime.Engine.GetService<IWorkflowDefinitionService>
();

//check if there's already a version of the WF
//!service.Workflows.Any(w => w.Id == workflowId)
if (true)
{
ImportDefinitions impDefinition = (ImportDefinitions)depService.CreateImportDefinitions(xml, package
Name, sharedResourcesPath, ImportType.Import);
impDefinition.SetPermissionMappings(new PNMsoft.Sequence.Deployment.Mappings.MemberMappingCollecti
on());
impDefinition.SetRecipientMappings(new PNMsoft.Sequence.Deployment.Mappings.MemberMappingCollectio
n());

//if you want to overwrite the existing one...
impDefinition.ImportType = ImportType.RestoreFrom;

depService.Import(impDefinition);

DiagnosticUtility.DesignTime.TraceInformationEvent("{0} Was Imported.", packageName);
Console.WriteLine("{0} Was Imported.", packageName);
}
else
{
DiagnosticUtility.DesignTime.TraceInformationEvent("{0} already exists.", packageName);
Console.WriteLine("{0} already exists.", packageName);
}
}

DiagnosticUtility.DesignTime.TraceInformationEvent("Finished importing WFs successfully");
Console.WriteLine(@"Finished packages import. Check the SVC log in C:\Program Files\PNMsoft\Shar
ed Resources\ImportLogs");
Console.ReadLine();
}
else
{
DiagnosticUtility.DesignTime.TraceInformationEvent("No Wfs found in the directory");
Console.WriteLine("No Wfs found in the directory");
}
}

}
catch (Exception ex)
{
DiagnosticUtility.DesignTime.TraceErrorData(ex);
Console.WriteLine(ex.Message);
Console.ReadLine();
}
}

internal static string[] GetFiles(string path)
{
string[] packages = Directory.GetFiles(Directory.GetCurrentDirectory() + path, "*.*.zip");
}

```

```
return packages;
return packages;
}
}
}
```