Data import instructions

This document explains the usage of the DataImport.ps1 script.

# Prerequisites

To use the script it is required a basic knowledge of PowerShell.

## PowerShell

The required version of **PowerShell (2.0)** is included with latest Windows versions. If it’s not on your system, please download it here: <http://support.microsoft.com/kb/968930/>

## Setup PowerShell to use .net 4

The computer must be configured to allow script execution and to use .net 4.0. Please check the topics “Running scripts with Windows PowerShell” and “Running PowerShell under .net 4.0” on the wiki page <https://tfsgorba.gorba.com/sites/teamsoftware/Wiki/Windows%20PowerShell.aspx>.

## Required files

To import data you need 6 files:

* ImportData.ps1 (the script)
* ImportDataUtils.psm1
* ValidateXml.ps1
* DocumentFormat.OpenXml.dll
* ClosedXML.dll
* Data excel file (by default, we use “DevelopmentSetup.xlsx”)
* (ExportItcsConfigurations.ps1 if existing VdvClient configurations must be imported)

You can find these files on the source repository of TFS:

* $/Gorba/Main/Center/BackgroundSystem/Tools/DataImporter/ImportData.ps1
* $/Gorba/Main/Center/BackgroundSystem/Tools/DataImporter/ImportDataUtils.psm1
* $/Gorba/Main/Center/BackgroundSystem/Tools/DataImporter/ValidateXml.ps1
* $/Gorba/Main/Center/BackgroundSystem/Tools/DataImporter/DocumentFormat.OpenXml.dll
* $/Gorba/Main/Center/BackgroundSystem/Tools/DataImporter/ClosedXML.dll
* $/Gorba/Main/Center/BackgroundSystem/Tools/DataImporter/ExportItcsConfigurations.ps1
* $/Gorba/Main/Center/BackgroundSystem/Deploy/Source/DevelopmentSetup.xlsx

## CenterCmdlets module

It is needed to have the CenterCmdlets module installed. There are two possibilities to install the module:

* To make it available to all users, copy the build output of the project BackgroundSystem.Powershell to “C:\Windows\System32\WindowsPowerShell\v1.0\Modules\CenterCmdlets”
* To make it available to a single user, copy the build output of the project BackgroundSystem.Powershell to “C:\Users\{user}\Documents\WindowsPowerShell\Modules\CenterCmdlets

# Execution

Before running the script, ensure that 7-zip is installed in your machine, in general at “C:\Program Files (x86)\7-Zip\7z.exe”. if not, you can download it and install it free of charge.

To execute the script, you have to open a PowerShell prompt.

There, you have to call the following commands:

* Move to the directory containing the script (using cd or pushd)
* Call the ImportData.ps1 script with parameters

Typically:

*pushd Path\_to\_scriptcontainingdirectory*

*ImportData.ps1 –ExcelFileName “absolute\_path\_to\_excel\_file” –ServicesBaseAddress “path\_to\_BackgroundSystem”*

Parameters:

* ExcelFileName: path to the excel file. It can be an absolute path or a path relative to the script folder. By default, it points to the excel file on the repository
* ServicesBaseAddress: base address where BackgroundSystem services can be found

Example:

*pushd c:\Gorba\Main\Center\Host\Tools\DataImporter*

*.\ImportData.ps1 –ServicesBaseAddress http://192.168.1.158/BackgroundSystem*

This example uses the default excel data file as found on the repository and uses services on a test machine (for instance the alarm service will be <http://192.168.1.158/BackgroundSystem/AlarmService.svc>)

WARNING: The script drops the database on the system and recreates it. All data are lost

# Know issues

* When you execute the script, all assemblies will be locked. You have to close the session to remove those “locks”.
* You must ensure that data in the excel file are consistent. With wrong data, you should “expect unexpected behaviors” ☺
* We use basic http binding which has by default very low limits in allowed message size. When you get data, please provide a low Take value to the filter.