

User manual

QSAR TOOLBOX

The OECD QSAR Toolbox
for Grouping Chemicals
into Categories

Toolbox addin container

Document history

Version	Comment
Version 1.0	October 2019: Toolbox addin container
Version 1.1	February 2021: Updated

Issue date: October 2019

Language: English

If you have questions or comments that relate to this document, please send them to ehscont@oecd.org or visit the QSAR Toolbox discussion forum at https://community.oecd.org/community/toolbox_forum

Toolbox addin container

Table of Contents

Document history 1

Issue date: 2

Language:..... 2

1 Introduction 4

2 Background 4

3 Transport container file structure..... 4

4 Configuration 8

Toolbox addin container

1 Introduction

This document provides a description of the QSAR Toolbox Addin container format. To provide some context some overview is provided on Toolbox internals. The container described in this document is used by the Toolbox repository website for distribution of Toolbox addins.

More information on the internals of the Toolbox docking infrastructure is available in the Toolbox Software Development Kit (SDK).

2 Background

The Toolbox can have third part modules(addins) docked via a public API. Once the module developer have prepared the module they need to prepare a single file package, that will be used for the Toolbox store and will contain all resources needed for the deployment of a working addin to Toolbox.

Toolbox addins infrastructure distinguishes between two separate entities:

1. **Toolbox Addin** - A C# assembly/module that implements *IToolboxAddin* (which provides a list of *IObjectFactory* Toolbox objects).
2. **Toolbox Instance (*IObjectFactory*)** - The Toolbox Instance furthermore comes in two sub-types.
 - Prepared Instance - *IObjectFactory* constructed by the Addin module
 - Configured instance – this is a particular Toolbox instance that is a product of a separate configuration file that translates to a single *IObjectFactory*

The Toolbox repository website would have to address the following use cases:

1. User downloading an **Addin** module
2. User downloading separate **Configured instance**. Each configuration should have a dependency on the Addin module version – updating an Addin may mean having to update relevant Configured instance(s).

3 Transport container file structure

The Toolbox Addin sub-system distinguishes three folders from which files will be deployed:

Toolbox addin container

- **[ToolboxServerAddinsFolder]** – the folder where the assemblies for the Addin are placed.
- **[ToolboxServerAddinConfigFolder]** – the folder where the Configured instance(s) are deployed
- **[ToolboxClientAddinsFolder]** – the folder where client side assemblies are placed

These placeholders are unique for each plugin and are set to a local file system folder upon addin deployment.

Additionally there is a folders from which files will not be deployed but rather containing executable to be called during deployment. Its purpose is to display per addin EULA, perform additional setup, etc.:

- **[DeploymentAdditional]**

Reflecting the deployment requirements detailed above the transport container has the following constraints:

1. Extension – the **Addin**'s file extension is ***.tbAddin**
2. Compression – the **Addin** is a ZIP archived package containing the **[ToolboxServerAddinsFolder]**, **[ToolboxClientAddinsFolder]**, **[ToolboxServerAddinConfigFolder]**, **[DeploymentAdditional]**, as well as a manifest file. Figure 1 shows how a folder structure looks like.

Name	Date modified	Type	Size
DeploymentAdditional	10/9/2019 8:40 AM	File folder	
ToolboxClientAddinsFolder	10/9/2019 8:39 AM	File folder	
ToolboxServerAddinConfigFolder	10/7/2019 1:43 PM	File folder	
ToolboxServerAddinsFolder	10/9/2019 8:40 AM	File folder	
manifest.txt	9/16/2019 3:35 PM	Text Document	1 KB

Figure 1: The folder structure within the **Addin** container

3. Manifest – the package should have a manifest file specifying containing one or more **NAME=Value** pairs. The name of the file: **"manifest.txt"**.

Recognized NAME items are:

- **Plugin metadata each**

Toolbox addin container

NAME	Description
ConfiguredInstanceDependentPlugin	[Required (for Configured instance)] The name:version for the plugin that the configured instance will be loaded by.
AddinAssemblyName	[Required (for Addin)] Addin assembly's filename. The value for AddinAssemblyName should be an existing assembly from the [ToolboxServerAddinsFolder] folder. For packages with configured instances only the AddinAssemblyName is the value used to determine which module will be responsible for constructing the Configured Instance from the file. E.g. <i>AddinAssemblyName = LMC.Toolbox.Server.Profiling.dll</i>
Guid	[Required] Globally Unique Identifier
ShortName	[Required] Global unique sort name for the plugin. Should not contain spaces.
LongName	[Required] Display name for the plugin
Version	[Required] Version of the plugin (major.minor)
Description	
Changelog	
PluginDependencies	Enumeration of the plugins dependencies in the form of <i>ShortName:Version</i> , in multiple dependencies are separated using semicolon ";" <i>Profiler-plugin:1.0</i>
URL	URL, if available, for the plugin

Toolbox addin container

- 4. Dependencies – the included assembly dependencies should be as minimal as possible. If dependencies are available in the Toolbox Server’s binaries folder then the dependency should not be present in the **[ToolboxServerAddinsFolder]** or **[ToolboxClientAddinsFolder]** folders.
- 5. **[DeploymentAdditional]** – this folder should contain an executable **DeploymentAdditional.exe**. During deployment it will be called by the deployment system with one or three parameters. In order:

- o ClientInstallDir – where the client files are deployed on the local file system.

-ClientInstallDir=C:\Program Files (x86)\QSAR Toolbox\QSAR Toolbox 4.4\Toolbox Client\Bin\Addins\LMC.Toolbox.Server.Profiling

- o ServerInstallDir – where the server addin files are deployed on the local file system.

-ServerInstallDir=C:\Program Files (x86)\QSAR Toolbox\QSAR Toolbox 4.4\Toolbox Server\Bin\Addins\LMC.Toolbox.Server.Profiling

- o ConfigInstallDir – where the config files are deployed on the local filesystem.

-ConfigInstallDir=C:\Program Files (x86)\Common Files\QSAR Toolbox 4.4\Config\Addins\LMC.Toolbox.Server.Profiling

In case of Toolbox installation “Desktop Client Only”, only first parameter will be provided.

The application will be called after the files from the **[ToolboxServerAddinsFolder]**, **[ToolboxClientAddinsFolder]**, **[ToolboxServerAddinConfigFolder]** folders have been copied to the local filesystem.

The application should exit with ExitCode **0** to indicate success. All other exit codes will mean the deployment procedure will be rolled back.

Toolbox addin container

If the **DeploymentAdditional.exe** creates persisting resources, such as registry values, or additional files, there must be an accompanying **DeploymentAdditionalUninstall.exe** that removes these resources. It will be called during addin uninstall.

There are two example plugins:

1. [Example.Plugin.ConfigurableInstance.tbaddin](#)

This plugin shows a deployment of a profiler instance, that uses the default Toolbox profiler module. It is basically a single config file (with the **Documentation** and **Reference** folder being optional).

2. [Example.Plugin.WithDeploymentAdditional.tbaddin](#)

This plugin demonstrates a full deployment of binary files, complete with a DeploymentAdditional functionalities.

LMC.Toolbox.Server.Profiling.tbAddin

4 Configuration

UI for downloading/updating is distributed with the QSAR Toolbox server. The application is named **ToolboxRepositoryClient**.

The online repository for distributing the addins is located at:

<https://repository.qsartoolbox.org/>

OECD

2, rue André Pascal

75775 Paris Cedex 16

France

Tel.: +33 1 45 24 82 00

Fax: +33 1 45 24 85 00

ehscont@oecd.org