



PDM Workbench NX

PDM Workbench NX Release 13.0 for Aras Innovator

Installation & Administration Manual

Version 1



Copyright

© 2005-2019 T-Systems International GmbH.

All rights reserved. Printed in Germany.

Contact

T-Systems International GmbH
Business Area PLM
Fasanenweg 5
70771 Leinfelden-Echterdingen
Germany

<http://plm.t-systems-service.com/en/pdm-workbench-nx>

☎ +49 (0) 40 30600 5544

✉ +49 (0) 3915 80125688

mail: cmi_support@t-systems.com

Manual History

Version	Date
1.2.5	September 2016
2.1.0	April 2017
2.2.0	Oktober 2017
2.3.1	July 2018
9.0	February 2019
10.0	May 2019
11.0	November 2019
12.0	May 2020
13.0	November 2020

This edition 13.0 obsoletes all previous editions.

Your Comments are Welcome

Please feel free to tell us your opinion; we are always interested in improving our publications. Mail your comments to:

T-Systems International GmbH
Business Area PLM
Fasanenweg 5
70771 Leinfelden-Echterdingen
Germany

mail: cmi_support@t-systems.com

Preface

About this Manual

This manual provides installation and configuration information for the PDM Workbench NX. Before using this guide, be sure you understand:

- the Microsoft Windows operating system
- the administration of the Siemens NX system
- the administration of the Aras Innovator system

Related Documents

The following manuals contain information about installation, administration, usage and customization of the PDM Workbench NX:

Manual Title	Version
<i>PDM Workbench NX Installation & Administration Manual</i>	13.0
<i>PDM Workbench NX User Manual</i>	13.0

Trademarks

NX is a registered trademark of Siemens Digital Industries Software.

Aras and Aras Innovator are registered trademarks of Aras Corporation.

Names of other products mentioned in this manual are used for identification purpose only and may be trademarks of their companies.

Table of Contents

CHAPTER 1	1
OVERVIEW	1
SYSTEM HARDWARE AND SOFTWARE REQUIREMENTS.....	1
<i>Aras Server</i>	1
<i>NX Client</i>	1
INSTALLATION STEPS	2
CHAPTER 2	3
ADAPTING NX	3
LOADING PWB NX SOFTWARE FROM CD-ROM	3
PWB NX INSTALLATION.....	3
<i>Configuring the installation</i>	3
SILENT INSTALLATION	11
<i>Parameters</i>	11
<i>Usage</i>	12
REQUIRED NX OPTIONS	12
LICENSE MANAGER INSTALLATION	13
TROUBLESHOOTING	13
TESTING THE INSTALLATION.....	13
SETTING OF ENVIRONMENT VARIABLES.....	14
CHAPTER 3	15
PDM WORKBENCH NX DATA MODEL	15
INSTALLATION.....	15
CHAPTER 4	17
PDM WORKBENCH SERVER DLL	17
COPYING THE DLL.....	17
MODIFYING THE SERVER CONFIGURATION FILE.....	17
CHAPTER 5	19
SERVER CONFIGURATION	19
CONFIGURATION VARIABLES	19
CONFIGURATION ITEMS.....	19
CHAPTER 6	21
CONFIGURATIONS FOR SPECIFIC FUNCTIONALITIES	21
STANDARD CONFIGURATION	21
<i>Exchange map</i>	21
<i>SOAP target URL</i>	21
<i>Session settings</i>	21
<i>Create Part mode</i>	22
<i>Key attribute</i>	22
<i>Class attribute</i>	22
<i>Relation attribute</i>	22
<i>Relationship attribute</i>	22
<i>Left relationship attribute</i>	22
<i>Right relationship attribute</i>	22
<i>Left relation class attribute</i>	22
<i>Right relation class attribute</i>	22
<i>Extended relation class attribute</i>	22
<i>Last modification date attribute</i>	22

DATA MODEL CONFIGURATION	23
<i>Document data model</i>	23
QUERY CONFIGURATION.....	23
<i>The Query dialog attributes</i>	23
<i>QueryOrderByAttribute</i>	23
<i>MaxQueryResults</i>	24
AUTONAME SUPPORT USING ARAS INNOVATOR SEQUENCE ITEMS.....	24
<i>Autoname functionality can use a server method</i>	25
OPEN IN ARAS IN NX CLIENT.....	28
CHAPTER 7	31
CLIENT SCHEMA FILE CONFIGURATION	31
STRUCTURE OF THE SCHEMA FILE.....	31
<i>Attributes of the tag "PWBSchema":</i>	31
<i>Display Names</i>	32
<i>Configuration settings</i>	32
<i>"object": 1 - n</i>	32
<i>"attribute": 0 - n</i>	32
<i>"pwbAttribute": 0 - n</i>	33
<i>"dataSource": 0 - n</i>	33
PDM ATTRIBUTES AND FORM ATTRIBUTES	33
<i>Description of the Widget Types</i>	34
<i>Login Form</i>	35
PDM OBJECTS.....	36
<i>Description of PDM Objects</i>	36
<i>Actions on PDM Objects</i>	37
<i>PDM Object Forms</i>	37
DATA SOURCES.....	38
<i>Data Source "Value" tag</i>	38
<i>Complete example of using a data source tag:</i>	39
CUSTOMIZING PDM WORKBENCH NX MENU	39

Table of Figures

PICTURE 1: DIRECTORY STRUCTURE OF THE PDM WORKBENCH INSTALLATION FILES	4
PICTURE 2: WELCOME TO THE INSTALLATION.....	5
PICTURE 3: LICENSE AGREEMENT	5
PICTURE 4: CHOOSE USERS.....	6
PICTURE 5: CHOOSE LOCATION OF PDM PACKAGE.....	6
PICTURE 6: CHOOSE LOCATION OF PDM PACKAGE (WITH PROPOSAL)	7
PICTURE 7: CHOOSE INSTALL LOCATION	7
PICTURE 8: CHOOSE NX INSTALLATION.....	8
PICTURE 9: CHOOSE EXCHANGE DIRECTORY	8
PICTURE 10: CHOOSE LOCATION OF SOAP TARGET URL	9
PICTURE 11: CHOOSE DATABASE NAME	9
PICTURE 12: SUBSUMPTION.....	10
PICTURE 13: INSTALLATION PROGRESS	10
PICTURE 14: INSTALLATION FINISHED	11
PICTURE 15: PDM ARAS INNOVATOR IMPORT UTILITY	15
PICTURE 16: ARAS INNOVATOR SERVER CONFIGURATION VARIABLES	19
PICTURE 17: PWB CONFIGURATION ITEM IN ARAS INNOVATOR.....	20
PICTURE 18: DEFINE ATTRIBUTE MAPPING	20
PICTURE 19: SAMPLE USEBOMPARTSTRUCTURE CONFIGURATION	23
PICTURE 20: SAMPLE QUERYORDERBYATTRIBUTE CONFIGURATION	23
PICTURE 21: SAMPLE MAXQUERYRESULTS CONFIGURATION	24
PICTURE 22: SAMPLE SEQUENCE ITEM.....	24
PICTURE 23: SEQUENCE ITEMS USED IN EXAMPLE.....	24
PICTURE 24: FILE “DEEPLINKING.JS”	28
PICTURE 25: FILE “DEEPLINKING.JS” SECOND OPTION	28
PICTURE 26: EMPTY TAB	29
PICTURE 27: ARAS INNOVATOR WEB CLIENT WITH CLIENT URL	29
PICTURE 28: WINDOW TITLE “ARAS INNOVATOR”	29
PICTURE 29: PWB CONFIGURATION SETTING “ARASWINDOWTITLESUBSTRING”	30
PICTURE 30: SINGLE LINE EDITOR WIDGET, UPDATE MODE	34
PICTURE 31: SINGLE LINE EDITOR WIDGET, OUTPUT MODE	35
PICTURE 32: MULTI LINE EDITOR WIDGET, UPDATE MODE	35
PICTURE 33: COMBO BOX WIDGET, SELECT MODE.....	35
PICTURE 34: SINGLE CHECK BOX WIDGET, SELECT MODE.....	35
PICTURE 35: PASSWORD BOX WIDGET	35
PICTURE 36: URL WIDGET	35

CHAPTER 1

Overview

This chapter provides basic information about the installation of the *PDM Workbench NX*.

System Hardware and Software Requirements

Aras Server

Hardware

For the Aras Server hardware sizing please refer to Aras recommended hardware sizing document.

Please refer to <https://www.aras.com/support/documentation/>

and look for the Platform specifications document, e.g.

Aras Innovator 11.0 - Platform Specifications.pdf

Software

Aras Innovator 11

Server Installation of Aras Innovator 11.0 SP12 on the following operating systems:

- Windows Server 2012, Windows Server 2014, Windows Server 2016
- Detailed information see [Aras Documentation](#): “Aras Innovator 11 Platform Specifications”

Aras Innovator 12

Server Installation of Aras Innovator 12.0 on the following operating systems:

- Windows Server 2012, Windows Server 2014, Windows Server 2016
- Detailed information see [Aras Documentation](#): “Aras Innovator 12 Platform Specifications”

NX Client

Hardware

The T-Systems NX Aras Connector PDM Workbench does not introduce any additional requirements to the CAD Workstations. The CAD Workstation spec should be close to the proposed certified hardware spec as defined by Siemens PLM for NX.

Please refer to:

<https://www.plm.automation.siemens.com/global/de/support/certifications.html>
for Operation System and CPU recommendations.

Select the respective NX Release and Windows 7 or Windows 10 64 bit as Operating System.

To process large NX Product Structures the CAD Workstation should have at least 32 GB RAM. For fast processing a SSD of sufficient size (500 GB) is recommended.

Software

Aras Innovator 11

On the NX client computers .NET 4.5.2 for Aras Innovator 11.0 SP12 has to be installed.

Aras Innovator 12

On the NX client computers .NET 4.7.2 for Aras Innovator 12 has to be installed.

NX

NX Client Version 10.0, Version 11.0, and Version 12.0 on the following operating systems:

- Windows 7 (64 Bit)
- Windows 10 (64 Bit)

Installation steps

This section describes which PDM Workbench NX modules (client and server) need to be installed.

On the client and the server two steps need to be performed each:

- Client installation: NX Add-in (chapter 2)
- Client installation: License Manager (For the installation of “licman20” please refer to the *Licman 2.0 Installation Manual*.)
- Server installation: PDM Workbench NX data model and server methods (chapter 3)
- Server installation: PDM Workbench NX server DLL (chapter 4)

CHAPTER 2

Adapting NX

The PDM Workbench NX module provided by T-Systems International GmbH extends the NX functionality to communicate with the Aras Innovator PDM system.

The **PWBNX_Vxx_xx** module includes all of the supported platform data in a compressed file. Thus, you should choose an installation location for all NX clients.

In the following example sections it is supposed that the software will be installed within the directory **C:\Program Files\T-Systems\PWBNX_Vxx_xx_Aras_xx** on Windows but you can surely choose any other destination for the module.

Within the installation you will need to supply the PDM specific installation package. The file name follows the naming convention **PWBCAD_xx_Aras_xx.zip**.

The installation does three things:

1. It copies the software on the hard disk.
2. It connects the Aras Innovator-NX-Integration to the installed NX installation.
3. It configures some settings of the Aras Innovator-NX-Integration.

To simplify the installation you can use the provided **setup.exe**.

Loading PWBNX Software from CD-ROM

Windows 7

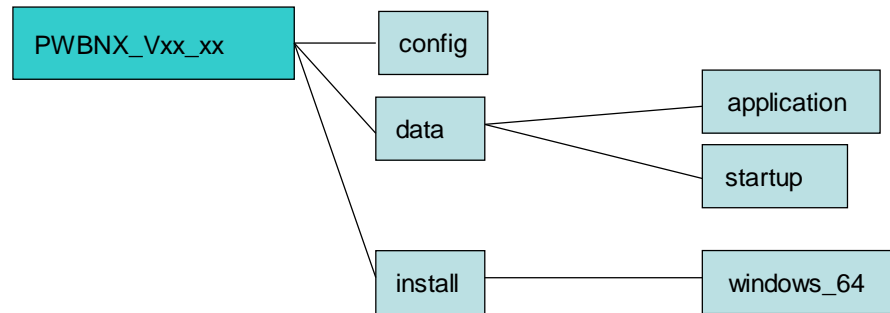
Use the Windows Explorer to locate the **D:\pwbpx\PWBNX_Vxx_xx.zip** file on the CD. Extract the content of the archive file to a temporary installation location.

PWBNX Installation

After you have successfully transferred the installation files to your installation host; the following steps will install the files and configure your installation.

Configuring the installation

The **PWBNX_Vxx_xx** Installation Directory has the following structure:



Picture 1: Directory structure of the PDM Workbench installation files

The **config** directory contains readme files and special files needed by the installer or the installed program.

The **data** directory contains the binary distributions for the PWB NX module.

The **install** directory contains the sub directory **windows_64** with all necessary data for the installer program.

Windows 7 (64 Bit)

On **Windows 7 (64 Bit)** use the Windows Explorer to run the **setup.exe** in the directory **PWBNX_Vxx_xx\install\windows_64** of the installation package if you have installed the 64 Bit version of NX.

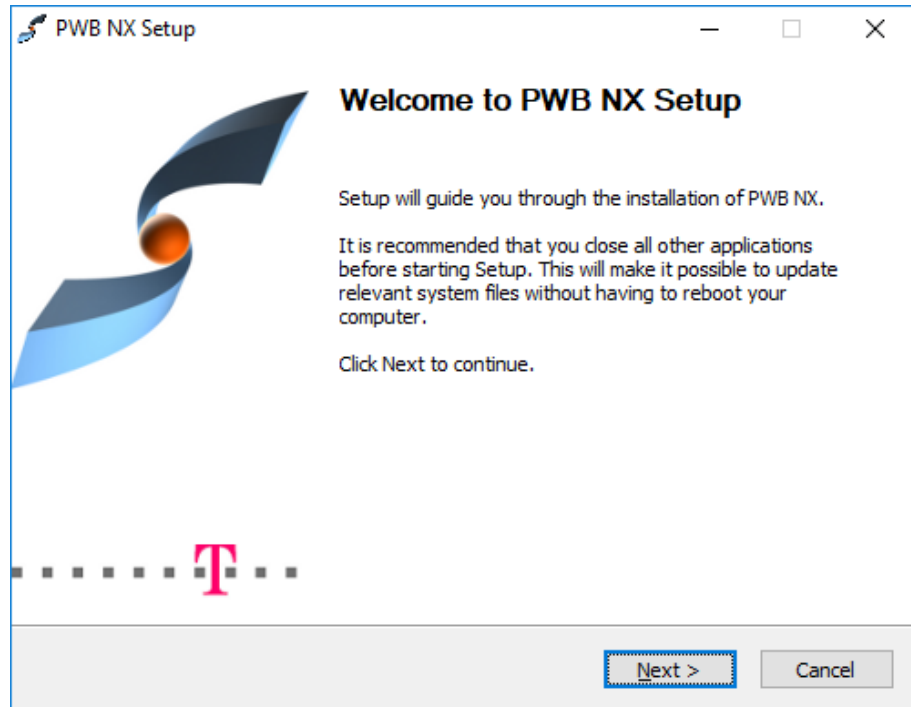
On **Windows Vista/Windows 7** the User Account Control (UAC) will be triggered and you will have to agree that the setup program may make changes to the computer. The installer is signed with a "**T-Systems International GmbH**" certificate to ensure its integrity and source.

The setup will **NOT** modify the native installation of NX.

The licman20 license manager has to be installed on the NX client host. For the installation of the license manager please refer to the *Licman 2.0 Installation Manual*.

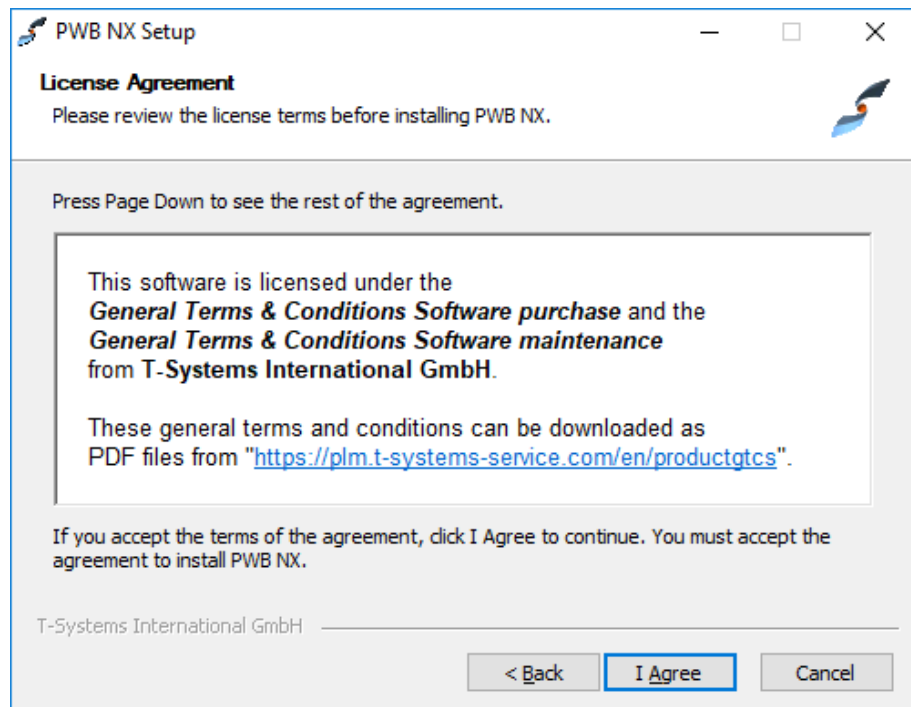
In the following the setup is shown step-by-step.

Installation process:



Picture 2: Welcome to the Installation

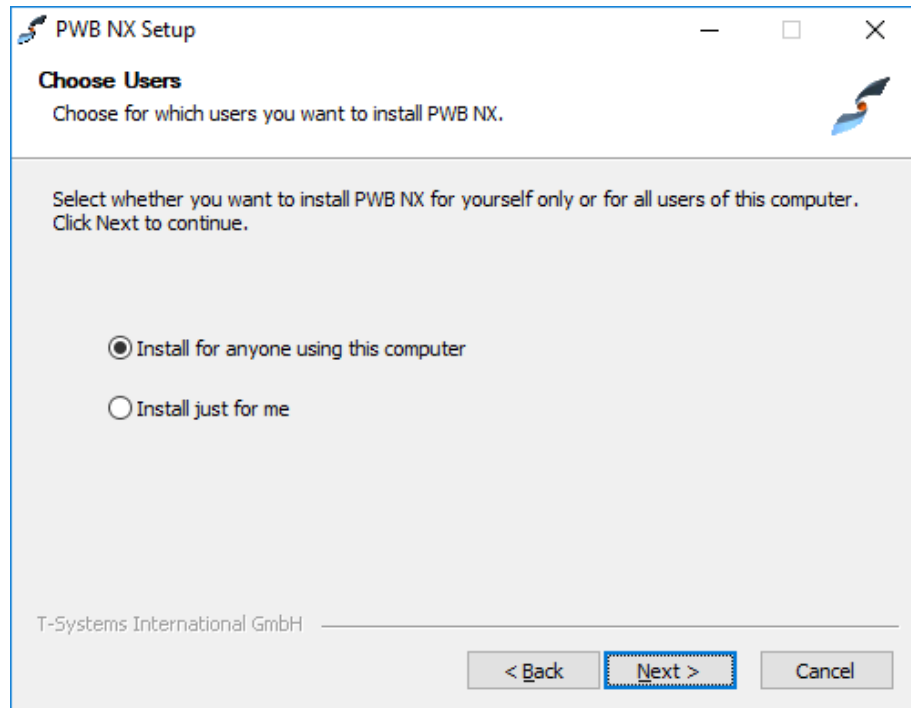
The installer software asks to approve the license terms (see *Picture 3: License Agreement*).



Picture 3: License Agreement

The installer software asks for the following input: **User scope**.

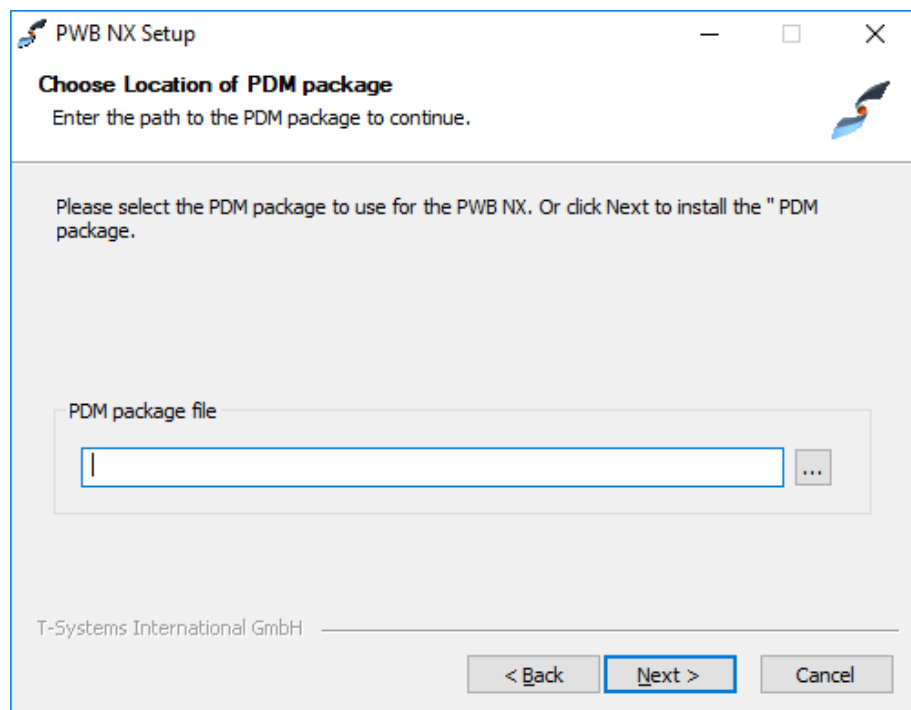
Next the installer will ask you to define the scope of the installation (see *Picture 4: Choose Users*). You can choose between an installation for anyone using the computer or just for the current user.



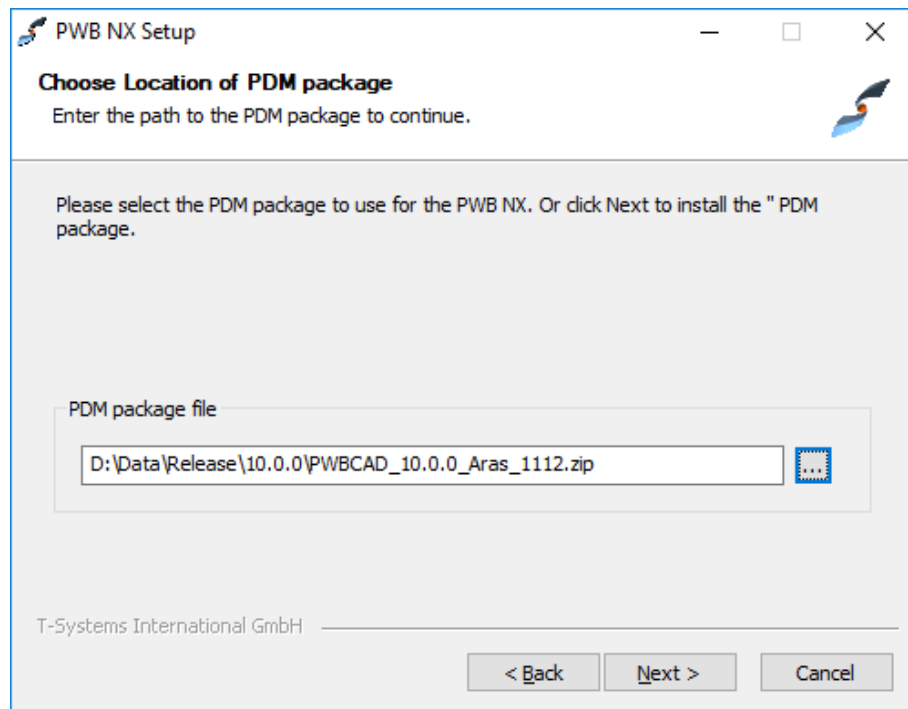
Picture 4: Choose Users

The installer asks for the following input: **Location of the PDM package.**

The installer asks for the location of the PDM package to use (see *Picture 5: Choose Location of PDM package*). If a PDM package has previously been unpacked within the installer it will be offered to install this package directly (see *Picture 6: Choose Location of PDM package (with proposal)*).



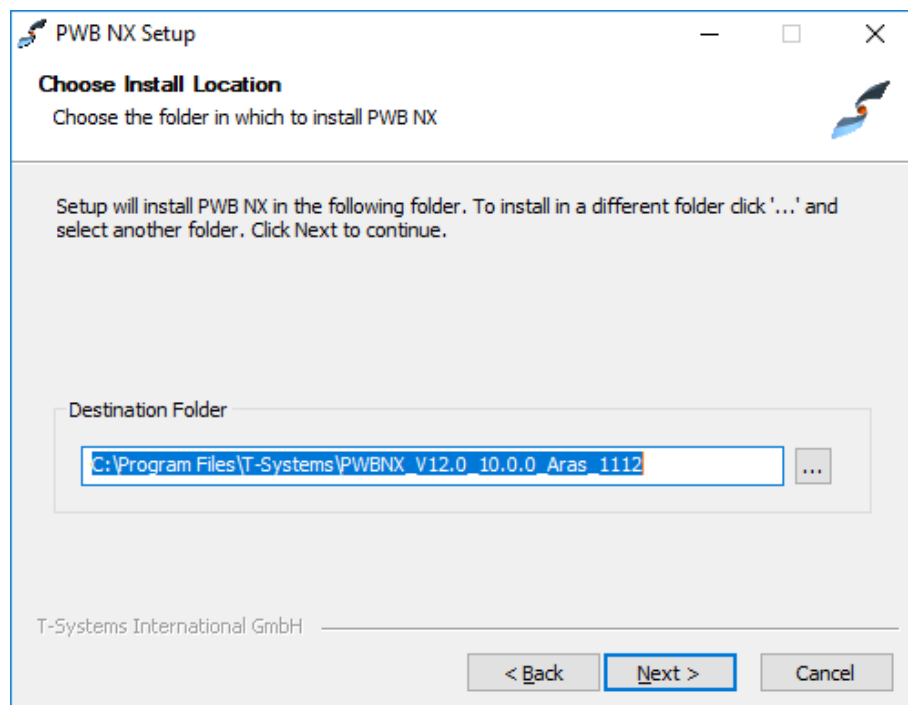
Picture 5: Choose Location of PDM package



Picture 6: Choose Location of PDM package (with proposal)

The installer software asks for the following input: **Installation directory**.

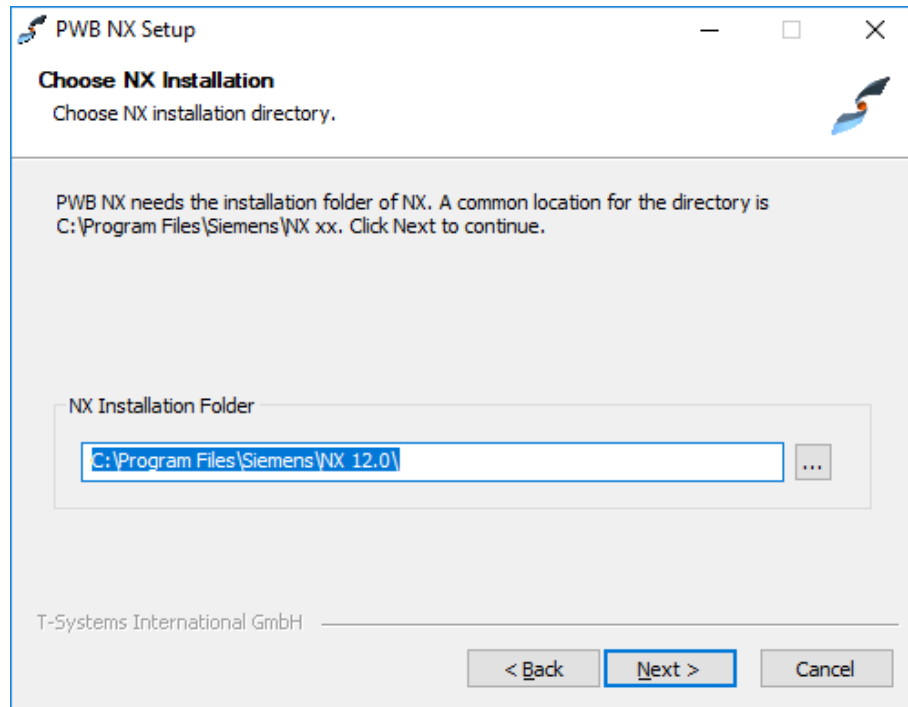
Next the installer will ask you for the target directory for the installation. You can use the given standard location or choose any other location (see *Picture 7: Choose Install Location*). The chosen folder must be empty or not exist.



Picture 7: Choose Install Location

The installer software asks for the following input: **NX installation directory**

The installation path of the NX to use needs to be specified (see *Picture 8: Choose NX Installation*).



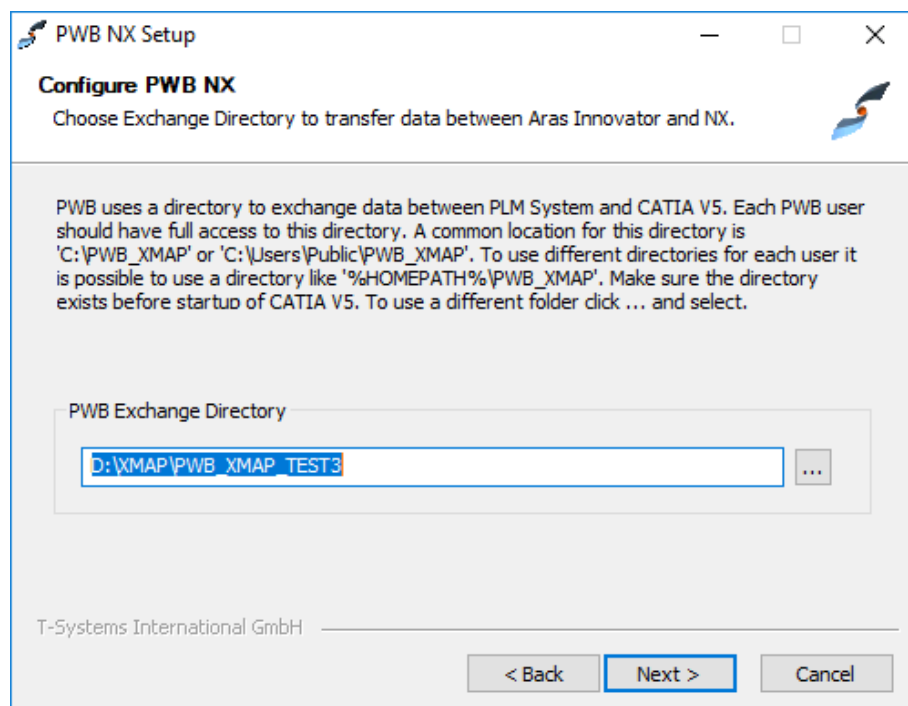
Picture 8: Choose NX Installation

The installer software asks for the following input: **PWB Exchange Directory**.

The PDM Workbench needs a temporary directory to perform the file transfer between NX and the PDM system. Make sure that this directory exists for every PDM Workbench NX user on the NX client machine.

You can either use the standard location or choose any other location (see *Picture 9: Choose Exchange Directory*).

If it is planned to run more than one NX session at a time each session must use its own PWB Exchange Directory!

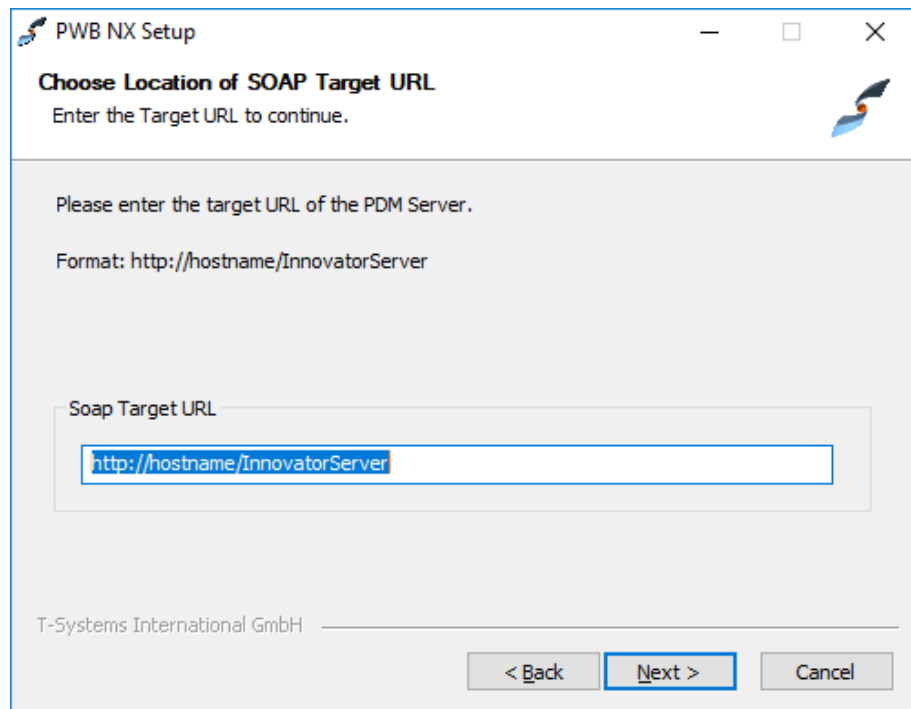


Picture 9: Choose Exchange Directory

The installer software asks for the following input: **SOAP Target URL**.

Finally you have to define the so called “Soap Target URL” for the PDM Server (see *Picture 10: Choose Location of SOAP Target URL*).

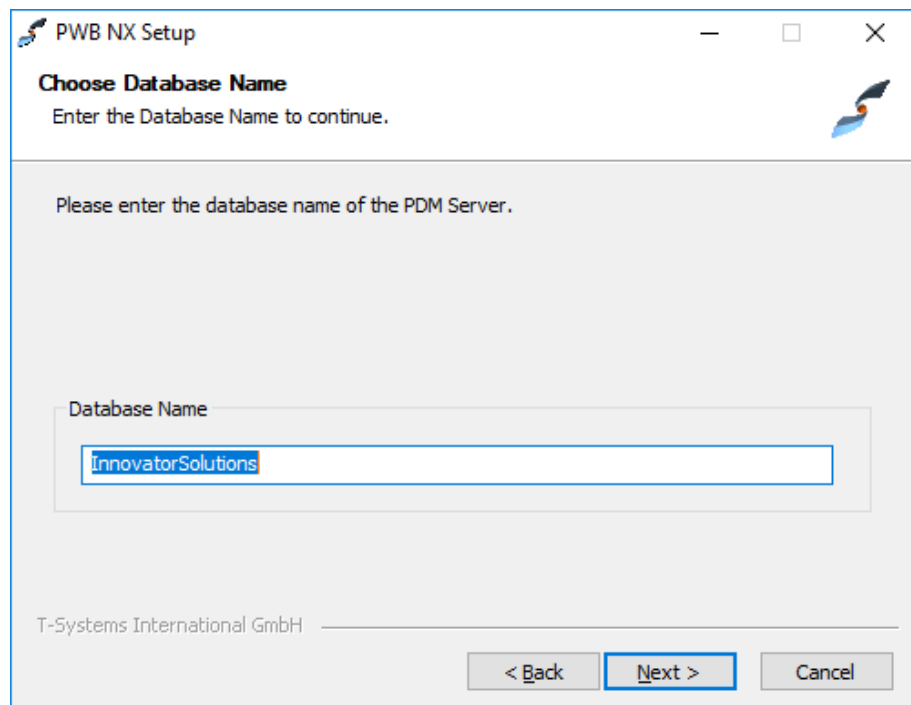
This URL defines the host on which the PDM Server is reachable.



Picture 10: Choose Location of SOAP Target URL

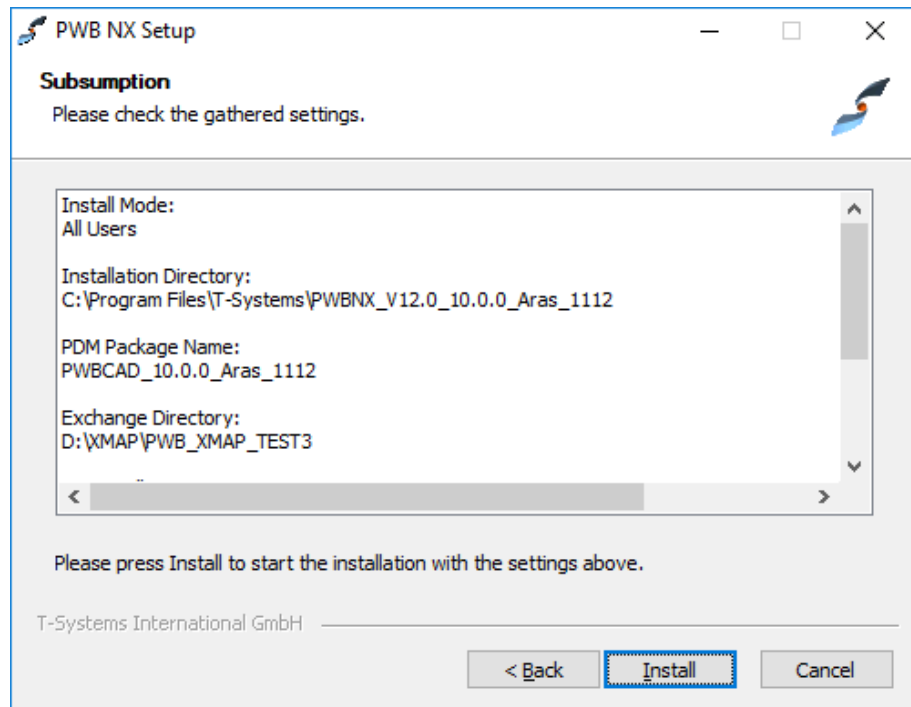
The installer software asks for the following input: **Database Name**.

Now you have to add the Database Name (see *Picture 11: Choose Database name*).



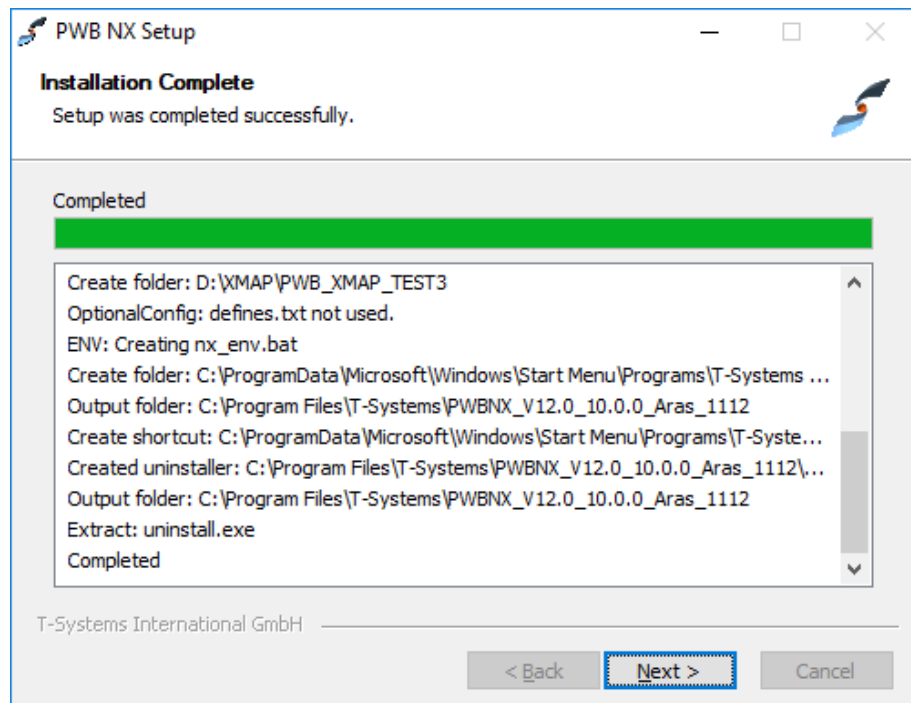
Picture 11: Choose Database name

After that you see the subsumption of your inputs before confirming them (see *Picture 12: Subsumption*).

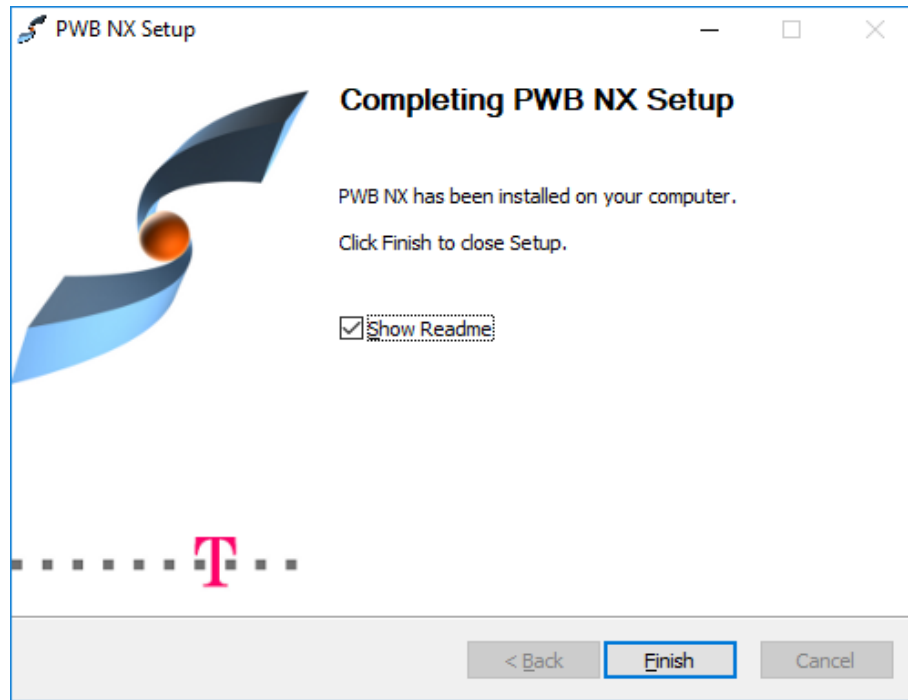


Picture 12: Subsumption

The installer will proceed in its process. The taken actions will be journalized (see *Picture 13: Installation progress*).



Picture 13: Installation progress



Picture 14: Installation finished

Silent Installation

It is possible to use a silent installation for the client installation.

Parameters

The following parameters are available for the silent installation:

Parameter name	Description	Sample value
/S	Activates the silent mode.	
/User= value	Installation only for yourself ("User") or for all users of the computer ("Admin"). Default is the highest possible value.	Admin
/PdmPackageNamePath= (File full path)	The full path of the zip file which includes the PDM package.	D:\aras_client\PWBCAD_10.0.0_Aras_1112.zip
/NxInstDir= (Directory path)	The directory of the NX installation.	C:\Program Files\Siemens\NX 12.0
/ExchangeMap= (Directory path)	The directory of the Exchange Directory.	C:\Users\Public\PWB_XMAP
/SoapTargetURL= (URL)	The SOAP target URL of the Aras server.	http://localhost/InnovatorServer
/DatabaseName= (Database Name)	The Database Name of the Aras server.	InnovatorSolutions
/D= (Directory path)	The target directory of the installation.	C:\Program Files\T-Systems\PWBNX_V12.0_10.0.0_Aras_1112

The parameters "/S" and "/SoapTargetURL" are required.

The parameter "/User" is optional. The highest possible value will be used as default value.

The parameter "/PdmPackageNamePath" is optional if the installation had run before. Then the last package will be used. If it is the first installation you have to provide the file path of the PDM package.

The parameter "/D" is optional. A part of the value will be taken from the current directory.

The value for the NX installation is optional; the value can be fetched from the Windows registry.

The parameter "/ExchangeMap" is optional. The directory "C:\Users\Public\PWB_XMAP" will be used as default value.

The parameter "/DatabaseName" is optional. The value "InnovatorSolutions" will be used as default value.

The parameter "/D" is optional. A part of the value will be taken from the current directory. It must be the last parameter used in the command line and must not contain any quotes, even if the path contains spaces. Only absolute paths are supported.

If one value is not given and it is not possible to fetch a value from the system the installation process will be stopped and the error message can be found in the file *install.log*.

Usage

For the silent installation please open a command line window as administrator.

Inside the temporary installation location, locate the folder "PWB_NX_Vxx_xx\install\windows_64" for an installation on a client with Windows 64 Bit.

Start the silent installation with a command line like this example:

```
Setup.exe /S /User= Admin /PdmPackageNamePath= D:\aras_client\
PWBCAD_10.0.0_Aras_1112.zip /NxInstDir= "C:\Program
Files\Siemens\NX 12.0" /ExchangeMap= "C:\Users\Public\PWB_XMAP"
/SoapTargetURL= "http://localhost/InnovatorServer"
/DatabaseName="InnovatorSolutions" /D= "C:\Program Files\T-
Systems\PWB_NX_V12.0_10.0.0_Aras_1112"
```

The log file of the installation will be stored in the current directory. There you can find the information about the installation process.

When the installation ended successful you will find the success message in this file.

Required NX Options

The following NX options are required:

- *File* → *Preferences* → *Assembly Load Options...* → *use partial loading* has to be switched off (NX 11), or
File → *Preferences* → *Assembly Load Options...* → *Scope / Option: Fully Load* should be set.
- *File* → *Utilities* → *Customer Defaults...* → *Gateway* → *Assembly Navigator* → *All* → *Pack Components* has to be switched off.

License Manager Installation

The licman20 license manager has to be installed on the NX client host. For the installation of the license manager please refer to the *Licman 2.0 Installation Manual*.

(For software and manual download see: <https://plm.t-systems-service.com/en/licman>)

Troubleshooting

Problem	Solution
"PDM Workbench" menu and/or toolbar not visible within NX.	Check the content of the NX log-file (Help-menu → "Log File"). There should be some lines with "TSI.PDM...". If you don't find such lines the connection between the Aras Innovator-NX-Integration and NX is not established.
Starting NX or a command from "PDM Workbench" results in a licensing error.	Check the existence of a license using the "Licman Test" available in Windows Start menu → T-Systems → Licman.
The login dialog shows not the correct Aras Innovator server address. The login dialog shows not the correct database.	Check the PDM Workbench NX Schema file, which is placed in the config directory of the installation. The right server address must be defined within the "soapTargetUri" XML element. The database must be defined within the "dataSource" XML element (<dataSource name="LoginDatabases" type="ValueList">)
The login fails.	Check the server address shown in the dialog. Check your credentials. In case of a server connection error: Check whether it is possible to connect to the Aras Innovator server with the Innovator thin client.

Testing the installation

For installation testing follow these steps:

1. Use: **Start→Programs→T-Systems PWB NX 12.0 10.0.0 Innovator 1112→pwb_nx_start** to launch NX.
2. Check that the menu "PDM Workbench NX" is visible within NX.
3. Press the "About" button in the "PDM Workbench NX" menu. The about dialog should be shown.

-
4. Press the “Login” button in the “PDM Workbench NX” menu. The login dialog should open and show the address of the Aras Innovator server entered during installation. Also the database from the installation must be selectable.

Once the steps 1. – 4. are successful the installation is ok.

For PDM Workbench NX functionality please refer to the *PDM Workbench NX User Manual*.

Setting of Environment Variables

The PDM Workbench software will use the following environment variables in the `NX` environment:

Environment variable	Comment
PWB_XMAP	The location of the exchange map directory. The exchange map directory must be unique for every started NX session on the same client.
PWB_SCHEMA_FILE	Path including file name of the XML configuration file.
PWB_SOAP_TARGET_URL	The URL of the web service. <code>http://localhost/InnovatorServer</code>
PWB_DATABASE_NAME	The name of the database, e.g. <code>InnovatorSolutions</code>
PWB_DEBUG	Set to “ON” to receive PWB debug output in the console.

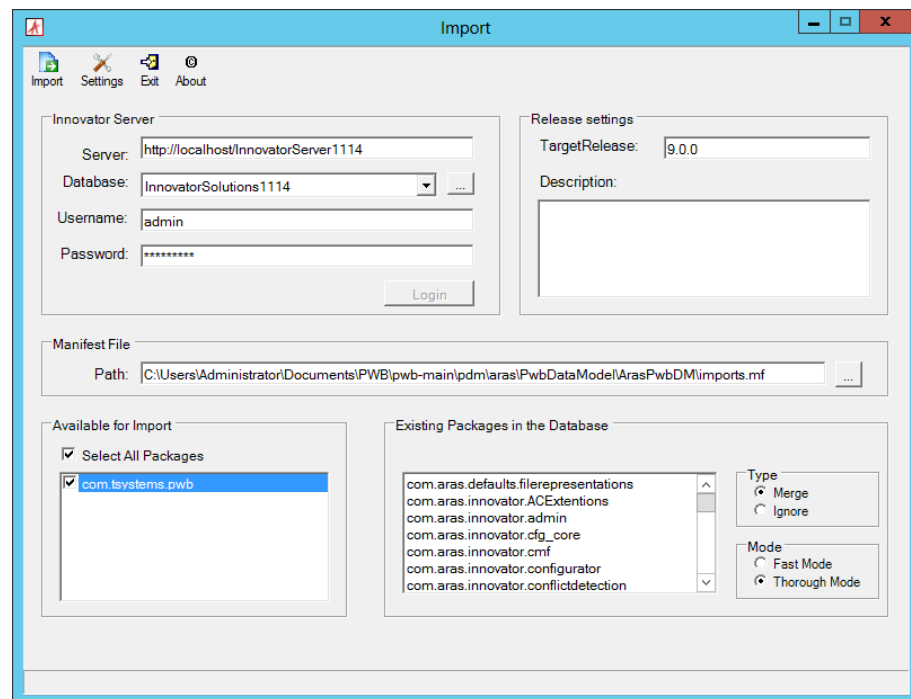
CHAPTER 3

PDM Workbench NX Data Model

Installation

The PDM Workbench NX data model and several server-side methods which call and support the main server functionality defined in the PDM Workbench NX server DLL (see *chapter 4*) need to be imported to Aras Innovator.

For this the “PwbDataModel_xx.zip” file needs to be unpacked first. Then four packages need to be imported to Aras Innovator with the Aras Innovator import utility¹:



Picture 15: PDM Aras Innovator import utility

Please select the manifest files

- PwbDataModel_10.0\ArasPwbDM\imports.mf
- PwbDataModel_10.0\ArasPwbDM_PLM\imports.mf
- PwbDataModel_10.0\ArasPwbDM_Core\imports.mf
- PwbDataModel_10.0\ArasPwbDM_NX_Additionals\imports.mf

¹ The import utility has to be downloaded from the Aras homepage and to be installed. Link: www.aras.com → Downloads → Download and Support → Additional Utilities → Package Import/Export Utility

in the import utility in this order and perform the import (Type is “Merge”, Mode is “Thorough Mode”) (see *Picture 15: PDM Aras Innovator import utility*).

CHAPTER 4

PDM Workbench Server DLL

Copying the DLL

Please copy the files

```
PwbServerAddin.dll
PwbServerAddin.pdb (optional)
```

from the distribution package to the Aras Innovator server directory

```
C:\Program Files\Aras\Innovator\Innovator\Server\bin
```

or to the corresponding directory if the Aras Innovator server has been installed in a different directory.

Modifying the server Configuration file

Also, please modify the file

```
C:\Program Files\Aras\Innovator\Innovator\Server\method-config.xml
```

by adding the highlighted lines:

```
...
<MethodConfig>
  <ReferencedAssemblies>
    <name>System.dll</name>
    <name>System.XML.dll</name>
    <name>System.Web.dll</name>
    <name>System.Data.dll</name>
    <name>System.Core.dll</name>
    <name>System.Configuration.dll</name>
    <name>System.Web.Extensions.dll</name>
    <name>$(binpath)/IOM.dll</name>
    <name>$(binpath)/InnovatorCore.dll</name>
    <name>$(binpath)/SPConnector.dll</name>
    <name>$(binpath)/ConversionManager.dll</name>
    <name>$(binpath)/FileExchangeService.dll</name>
    <name>$(binpath)/Conversion.Base.dll</name>
    <name>$(binpath)/Aras.TDF.Base.dll</name>
    <name>$(binpath)/Aras.ES.dll</name>
    <name>$(binpath)/PwbServerAddin.dll</name>
  </ReferencedAssemblies>
  ...
  ...
  <Template name="CSharp" line_number_offset="35">
    <![CDATA[using Aras.IOM;
using System;
```

```
using System.Collections;
using System.Collections.Generic;
using System.Data;
using System.Globalization;
using System.IO;
using System.Linq;
using System.Net;
using System.Text;
using System.Web;
using System.Web.SessionState;
using System.Xml;
using PwbServerAddin;

namespace $(pkgname)
{
    ...
}
```

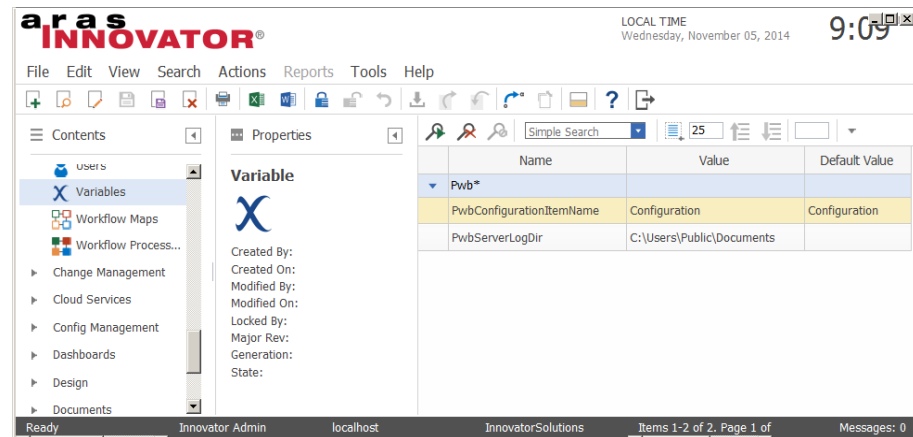
CHAPTER 5

Server Configuration

This chapter describes the configuration of the server-side of the PDM Workbench NX integration.

Configuration Variables

The following Aras Innovator server configuration variables need to be set for PDM Workbench NX to work correctly:



Picture 16: Aras Innovator server configuration variables

- PwbConfigurationItemName

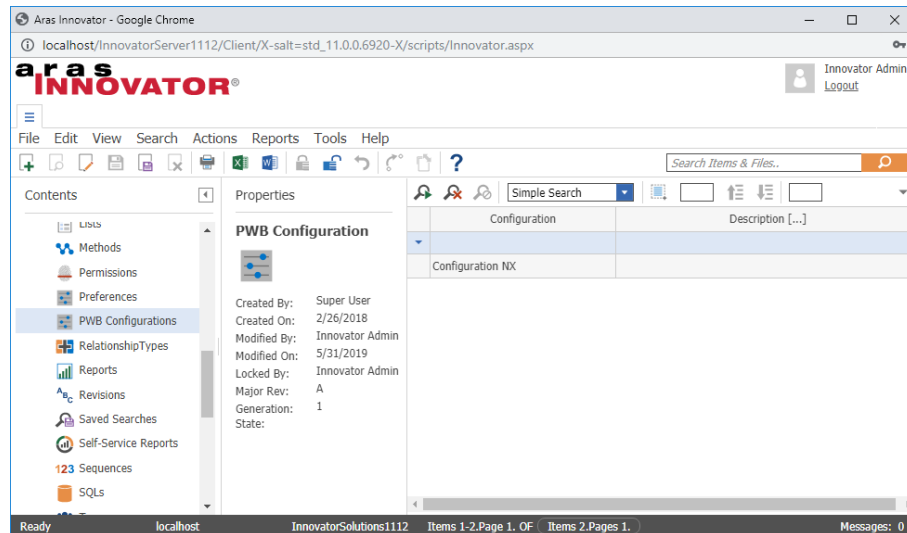
The name of the PDM Workbench configuration item which contains additional configuration information, like the attribute mapping configuration. Please see “Configuration Items” for more details.

- PwbServerLogDir

The absolute path of the directory into which the server log file should be written. If this variable is empty then no server log file will be written.

Configuration Items

In order to define the environment variables and to configure the mapping of attributes between Aras Innovator and NX a special configuration item (see *Picture 17: PWB Configuration item in Aras Innovator*) has to be used:



Picture 17: PWB Configuration item in Aras Innovator

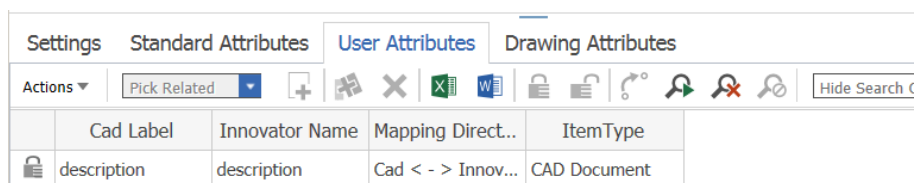
Open this item. The following configuration variables might be set:

- **ShowCreateDialogsDuringUpdate:** This configuration value manages the availability of the create dialog for new objects during update.
- **UseServerMethodesForAutoname:** The configuration determines, if sequences items or server methods are used for autonaming. (See *Autoname Support using Aras Innovator Sequence Items*)

To define the attribute mapping, open the configuration item and switch to the “User Attributes” view.

Enter a new entry for each attribute you want to map. Fill in the flowing values:

- **Cad Label:** the name of the attribute in NX
- **Innovator Name:** the name of the attribute in Aras Innovator
- **Mapping Direction:** The direction in which you want to map. From CAD System to PLM, the other way around or bidirectional.
- **ItemType:** Enter here for CAD Document, to map the attributes of the itemtype of CAD Documents. Parts are not supported at the moment.



Picture 18: Define Attribute Mapping

CHAPTER 6

Configurations for specific functionalities

This chapter describes the configuration of the PDM Workbench NX for specific functionalities.

Standard configuration

Exchange map

In the PDM Workbench NX Schema file the absolute path of the exchange map directory, where the downloaded NX files are stored, can be configured.

Example:

```
<xmap value="C:\PWB_XMAP" />
```

If the exchange map value is defined by the environment variable "PWB_XMAP", then that takes precedence. The definition in the Schema file takes effect only if such a NX environment variable does not exist.

Optional.

SOAP target URL

In the Schema file the URL of the server process, that the PDM Workbench NX client uses for its SOAP requests, can be configured.

Example:

```
<soapTargetUrl value="http://hostname/InnovatorServer" />
```

If the soap target URL value is defined by the environment variable "PWB_SOAP_TARGET_URL", then that takes precedence. The definition in the Schema file takes effect only if such a NX environment variable does not exist.

Optional.

Session settings

In the Schema file the session settings of the PDM Workbench NX can be defined. The following entries are supported.

Example:

```
<sessionSettings>  
    <passwordEncryption name="MD5" />  
</sessionSettings>
```

Optional.

Create Part mode

Depending on the value of the Schema file attribute createparts, Aras Innovator Part items are associated with each CAS Document item.

Example:

```
<createparts value="true"/>
```

Optional.

Key attribute

Internal attribute, do not change.

Class attribute

Internal attribute, do not change.

Relation attribute

Internal attribute, do not change.

Relationship attribute

Internal attribute, do not change.

Left relationship attribute

Internal attribute, do not change.

Right relationship attribute

Internal attribute, do not change.

Left relation class attribute

Internal attribute, do not change.

Right relation class attribute

Internal attribute, do not change.

Extended relation class attribute

Internal attribute, do not change.

Last modification date attribute

In the Schema file the name of the last modification date attribute can be defined.

Example:

```
<lastModificationDateAttribute name="last_mod_date" />
```

Default value: last_mod_date

Optional.

Data model configuration

The PWB Configuration item setting “UseBomPartStructure” indicates which data model will be used with the PDM Workbench NX.

The attribute has to be set to “false”, because the “BOM Part Structure Data Model” is not supported for the PDM Workbench NX.

UseBomPartStructure	false
---------------------	-------

Picture 19: Sample UseBomPartStructure configuration

Default value: false

Optional.

Document data model

In the “CAD Document Structure Data Model” the PDM structure is represented by CAD Documents. The relation “CAD Structure” is used.

Each CAD Document includes the NX file.

Query configuration

The Query dialog attributes

It is possible to configure the Query dialog. The Query dialog configuration is done with the “form name” tag “Query”.

Within the “form name” tag the “formAttribute” tags are defined.

```
<form name="Query">
  <formAttribute name="item_number"
    ...
  />
  ...
</form>
```

QueryOrderByAttribute

The PWB Configuration item setting “QueryOrderByAttribute” defines an attribute by which the query results are internally ordered. This is not noticeable by the user, but it can result in significant performance improvements when a query is performed if the attribute is in the database index.

QueryOrderByAttribute	id
-----------------------	----

Picture 20: Sample QueryOrderByAttribute configuration

Optional.

MaxQueryResults

The PWB Configuration item setting “MaxQueryResults” defines the maximum number of items that are retrieved in a single query.

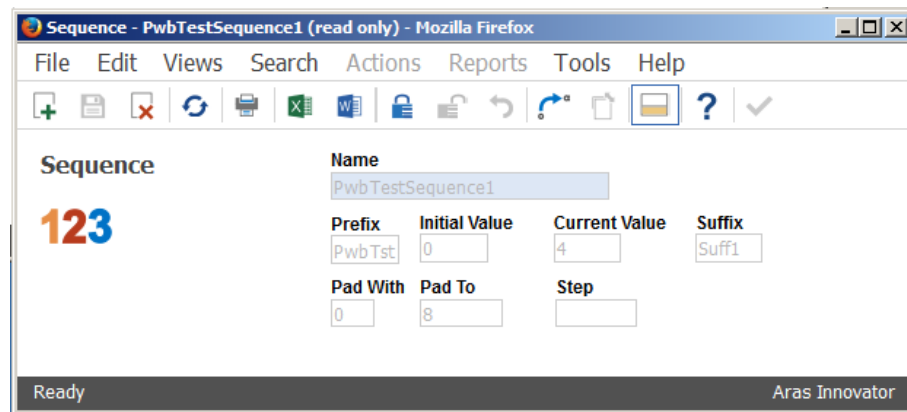
MaxQueryResults	100
-----------------	-----

Picture 21: Sample MaxQueryResults configuration

Optional.

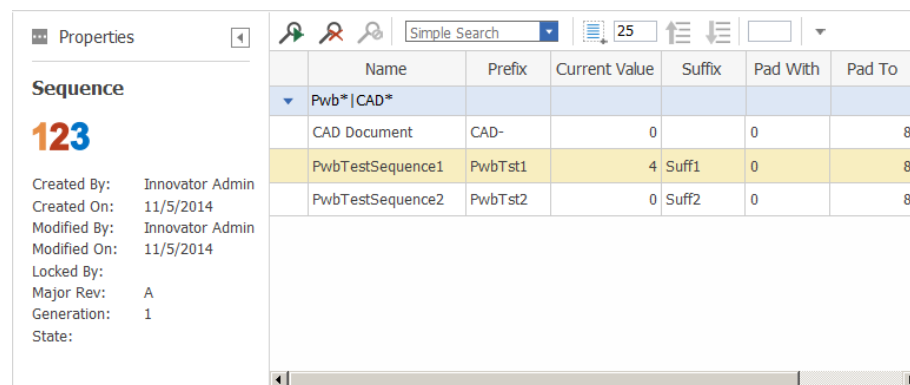
Autaname Support using Aras Innovator Sequence Items

First sequence items which should be used for the autaname functionality need to be created:



Picture 22: Sample Sequence item

The following sequence items will be used in the configuration example:



Name	Prefix	Current Value	Suffix	Pad With	Pad To
Pwb* CAD*					
CAD Document	CAD-	0		0	8
PwbTestSequence1	PwbTst1	4	Suff1	0	8
PwbTestSequence2	PwbTst2	0	Suff2	0	8

Picture 23: Sequence items used in example

Then the sequence items to be used need to be configured in the schema file:

First an attribute with a data source which contains the names of the sequence items needs to be defined:

```
<attribute name="pwbAutanameRule" displayName="NLS_AutanameRule" dataSource="AutanameRules" />
```

```
<dataSource name="AutanameRules" type="ValueList">
```



```

    <value name="" displayName="None" />
    <value name="PwbTestSequence1" displayName="" />
    <value name="PwbTestSequence2" displayName="" />
    <value name="CAD Document" displayName="" />
</dataSource>

```

Then a corresponding form attribute has to be included in the login dialog.

```

<form name="Login" info="ShowOnlyLoginData" >
  <frame displayName="NLS_UserData">
    ...
    <formAttribute name="pwbAutonameRule"
      displayName="Autoname Rule"
      widgetType="ComboBox" mode="update"
      visibleLength="15" required="false"
      entryAllowed="false" />
  </frame>
</form>

```

The setting **“UseServerMethodsForAutoname”** has to be set to **“false”** in the active PWB Configuration item.

UseServerMethodsForAutoname	false
-----------------------------	-------

This will enable the user to select a sequence item name as an autoname rule at login.

Autoname functionality can use a server method

The autoname functionality can use a server method instead of using a sequence item directly for obtaining a PDM-generated part or document number value.

The autoname functionality has to be configured in the file **“PWBSchema.xml”**.

The server method to be used needs to be configured in the Schema file:

First an attribute with a data source which contains the name of the server method needs to be defined:

```

<attribute name="pwbAutonameRule" displayName="NLS_AutonameRule"
  dataSource="AutonameRules" />

<dataSource name="AutonameRules" type="ValueList">
  <value name="PwbAutonameMethod1" displayName="" />
</dataSource>

```

Then a corresponding form attribute has to be included in the login dialog ...

```

<form name="Login" info="ShowOnlyLoginData" >
  <frame displayName="NLS_UserData">
    ...
    <formAttribute name="pwbAutonameRule"
      widgetType="ComboBox"
      mode="update" visibleLength="15"
      required="false" entryAllowed="false" />
  </frame>
</form>

```

... and in the “Set PDM Configuration” dialog.

```
<form name="PdmSessionConfig">
    <formAttribute name="pwbAutonameRule" widgetType="ComboBox"
        mode="update" visibleLength="15" required="false"
        listViewRelevant="true" />
</form>
```

This will enable the user to select a sequence item name as an autoname rule either at login or later while working in the PDM Workbench session.

In order for the “Set PDM Configuration” dialog to appear the setting “SetSessionConfig” has to be removed from the “removeToolbarIcons” definition:

```
<removeToolbarIcons>
    <!-- "LocalWorkspace", "Register", "Update", "Synchronize", "Refresh",
        "SetSessionConfig", "NewPwbWindow", "About"
    -->
    <!-- <icon name="LocalWorkspace" /> -->
    <icon name="Register" />
    <!-- <icon name="Update" /> -->
    <icon name="Synchronize" />
    <!-- <icon name="Refresh" /> -->
    <!-- <icon name="SetSessionConfig" /> -->
    <icon name="NewPwbWindow" />
    <!-- <icon name="About" /> -->
</removeToolbarIcons>
```

The setting “UseServerMethodsForAutoname” has to be set to “true” (default) in the active PWB Configuration item.

An additionally server method whose name corresponds to the name configured in the file “PWBSchema.xml” (e.g. “PwbAutonameMethod1”; “Server-Side”; “C#”; “Execution allowed to World”) has to be defined on the Aras Innovator server.

The server method can use information from standard NX attributes of the NX files to be imported to PDM, or values from PDM Workbench dialogs.

This is an example of such a server method:

```
// CC0.Utilities.WriteDebug("_PwbOutput", "Entering 'PwbAutonameMethod1' ");

var PwbServerApiObj = new PwbServerAddin.PwbServerApi(this);

// Preparing the input information
string Autoname = this.getProperty("Autoname");
string PdmType = this.getProperty("Type");
string PdmClassification = this.getProperty("Classification");

Item CadStdPropsItem = this.getPropertyItem("CadStdProps");
IDictionary<string, string> CadStdPropsDict = null;
if (CadStdPropsItem != null)
{
    CadStdPropsDict = PwbServerApiObj.DialogAttrsItemToDictionary(CadStdPropsItem);
}

Item CadDocInputDialogItem = this.getPropertyItem("CadDocDialogAttrs");

IDictionary<string, string> CadDocInputDialogDict = null;
if (CadDocInputDialogItem != null)
{
    CadDocInputDialogDict =
        PwbServerApiObj.DialogAttrsItemToDictionary(CadDocInputDialogItem);
}

Item PartInputDialogItem = this.getPropertyItem("PartDialogAttrs");
IDictionary<string, string> PartInputDialogDict = null;
```

```

if (PartInputDialogItem != null)
{
    PartInputDialogDict = PwbServerApiObj.DialogAttrsItemToDictionary(PartInputDialogItem);
}

// Place the custom algorithm here
string OutputLogInfo = "";

OutputLogInfo += "Autoname:" + Autoname + "" + "|";
OutputLogInfo += "PdmType:" + PdmType + "" + "|";
OutputLogInfo += "PdmClassification:" + PdmClassification + "" + "|";

if (CadStdPropsDict != null)
{
    OutputLogInfo += "CadStdProps:" + "|";

    var Enumerator = CadStdPropsDict.GetEnumerator();
    while (Enumerator.MoveNext() == true)
    {
        var CurrentVal = Enumerator.Current;
        OutputLogInfo += "" + CurrentVal.Key + "->" + CurrentVal.Value + "" + "|";
    }
}

if (CadDocInputDialogDict != null)
{
    OutputLogInfo += "CadDocInputDialog:" + "|";
    var Enumerator = CadDocInputDialogDict.GetEnumerator();
    while (Enumerator.MoveNext() == true)
    {
        var CurrentVal = Enumerator.Current;
        OutputLogInfo += "" + CurrentVal.Key + "->" + CurrentVal.Value + "" + "|";
    }
}

if (PartInputDialogDict != null)
{
    OutputLogInfo += "PartInputDialog:" + "|";

    var Enumerator = PartInputDialogDict.GetEnumerator();
    while (Enumerator.MoveNext() == true)
    {
        var CurrentVal = Enumerator.Current;
        OutputLogInfo += "" + CurrentVal.Key + "->" + CurrentVal.Value + "" + "|";
    }
}

// Getting the actual autoname value
string AutonameValue = "";
if (PdmType == "Part")
{
    AutonameValue = PwbServerApiObj.GetNextAutonameSequence("CAD Document");
}
else
{
    if ((PartInputDialogDict != null) &&
        (PartInputDialogDict.ContainsKey("item_number")))
    {
        AutonameValue = PartInputDialogDict["item_number"];
    }
    else
    {
        AutonameValue = PwbServerApiObj.GetNextAutonameSequence("CAD Document");
    }
}

OutputLogInfo += "done";

// Preparing the output
IDictionary<string, string> OutputInfoDict = new Dictionary<string, string>();

OutputInfoDict.Add("AutonameValue", AutonameValue);
OutputInfoDict.Add("LogLines", OutputLogInfo);

Item OutputInfoItem = PwbServerApiObj.DialogAttrsDictionaryToItem(OutputInfoDict);

```

```
return OutputInfoItem;
```

Open in Aras in NX Client

Single CAD items can be loaded in the Aras Innovator web client from NX.

Configuration

When using “Open in Aras” an additional “helper window” is displayed even if the session is already open. It is possible to close this window automatically after the Item is opened.

Modify the file

\Innovator\Client\Modules\aras.innovator.core.MainWindow\deepLinking.js in your Aras code tree. Around line 24 you can add an additional snippet of code to close the window if an Aras Innovator instance is already open.

```
if (!deepWindowIframe.src)
{
    deepWindowIframe.src = 'deepLinking.aspx';

    // Close this window if Innovator is already open
    window.close();
}
```

Picture 24: File “deepLinking.js”

According to your used Browser and security settings you will get a message like: “The webpage you are viewing is trying to close this window ... YES | NO”

Background: Strictly spoken java script is only allowed to close a window that was opened by the same script. As the window in case was opened by a system call (like double click a html file in the explorer) the script is not allowed to close the window.

To work around this and avoid the message you can use the following code instead “window.close()” in the file deeplinking.js:

```
if (!deepWindowIframe.src)
{
    deepWindowIframe.src = 'deepLinking.aspx';

    // Close this window if Innovator is already open
    var objWin = window.self;
    objWin.open('', '_self', '');
    objWin.close();
}
```

Picture 25: File “deepLinking.js” second option

Save the deepLinking.js file and restart the IIS.

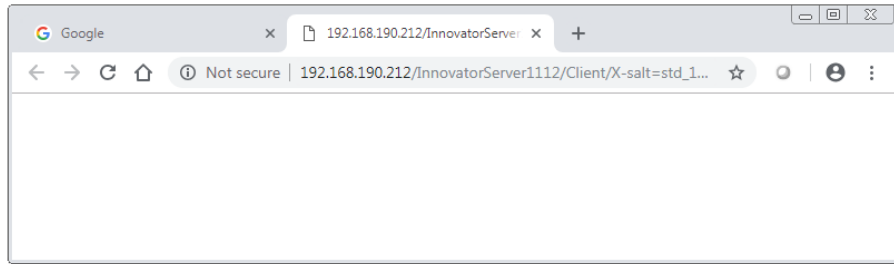
At all clients the browsers cache of the default browser must be cleared to enforce the use of the changed script.

Please note that this proposed change was tested in Aras Innovator 11.0 SP12, 11.0 SP15 and Aras 12.0 up to SP9.

For later Aras Innovator version this behavior may be changed.

For Aras Innovator 11 only

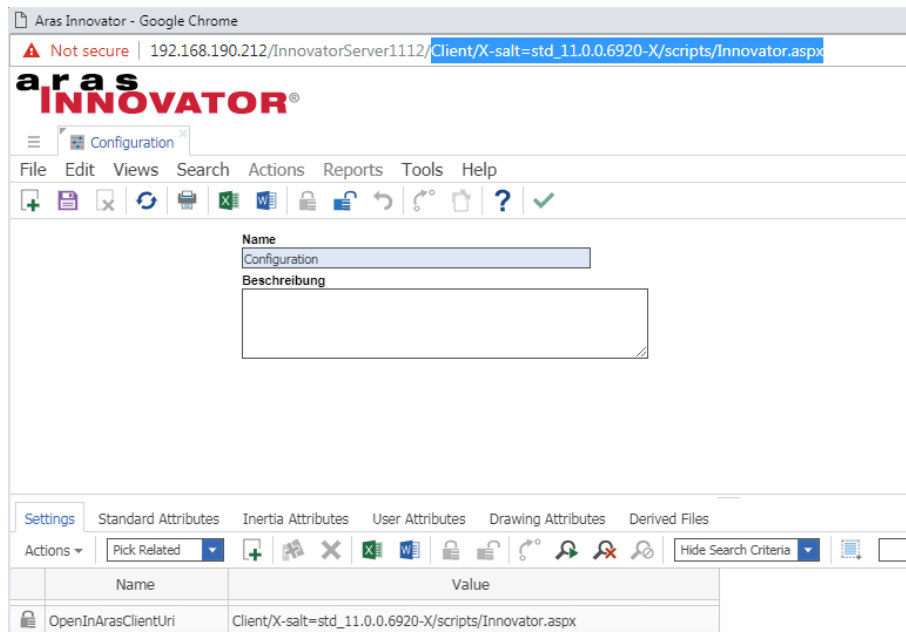
With Aras Innovator 11 the “Open In Aras” functionality opens a new empty tab in an additional default browser window.



Picture 26: Empty tab

To avoid this window, you must set the PWB Configuration “OpenInArasClientUri” to the client part of your Aras installation. You can obtain the value from the URL in the Aras Innovator Window:

http://192.168.190.212/InnovatorServer1112/Client/X-salt=std_11.0.0.6920-X/scripts/Innovator.aspx



Picture 27: Aras Innovator web client with Client URL

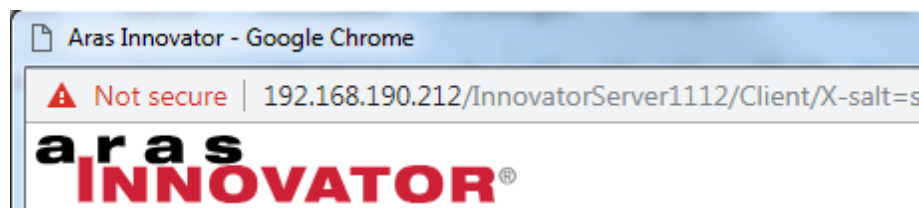
This setting must not exist in Aras Innovator 12.

To make sure the window handling works correctly an additional default browser window must exist (this could be the initial browser window of the Aras Innovator login).

Window Management

If you use “Open in Aras” this functionality brings the Aras Innovator window on top.

By default the Aras Innovator window uses “Aras Innovator ...” as window title:



Picture 28: Window title “Aras Innovator”

If your customization uses a different window title you have to set the PWB Configuration “ArasWindowTitleSubString” to your window title.

ArasWindowTitleSubString	Aras Innovator
--------------------------	----------------

Picture 29: PWB Configuration setting “ArasWindowTitleSubString”

CHAPTER 7

Client Schema File Configuration

This chapter describes the configuration of the client-side of the PDM Workbench NX integration.

Structure of the Schema File

The main purpose of the PDM Workbench NX Schema file is to define which subset of the objects, relations, and attributes in the PDM system should be made available to the design engineer who is working with NX and who needs to save the NX files he is working on in a PDM system.

The classes of PDM objects that the user can query, create, etc. will be defined in the Schema file, as well as the dialogs which contain these objects' attributes and the PDM relations which relate the PDM objects to each other.

The Schema file can be edited with a text editor, or a XML editor.

At the root of the Schema XML file, there is the tag *PWBSchemata*. Its child tags are named *PWBSchema*. The information about every PDM system that can be accessed is defined inside this *PWBSchema* tag. There is one *PWBSchema* tag for every PDM system and every PDM system customization that can be accessed from the PDM Workbench NX.

```
<!-- root tag -->
<PWBSchemata>
  <!-- out-of-the-box Aras -->
  <PWBSchema system="Aras" customization="Aras"
             displayName="NLS_System" visibleLength="15">
    ...
  </PWBSchema>

  <!-- customization of Aras -->
  <PWBSchema system="Aras" customization="PDM-Customization"
             displayName="NLS_System" visibleLength="15">
    ...
  </PWBSchema>
</PWBSchemata>
```

Attributes of the tag "PWBSchema":

- "system" Contains the short name of the PDM system. Supported is "Aras" for Aras Innovator.

- “customization” Contains the name of the customization. If the PDM system is used out of the box without any customization, then the convention is to use the short name as defined for the attribute *system*.
- “displayName” Contains the NLS (native language support) name of the PDM system or customization that is defined in the tag *PWBSchema*.
- “visibleLength” Contains the visible length of the display name to be shown in the dialogs of NX.
- “allowedLength” Contains the allowed length of the values inserted in the text editor widgets in characters.

Display Names

Many XML tags (*PWBSchema*, *frame*, *language*, *object*, *relation*, *attribute*, etc.) have an attribute with the name *displayName*. The string that represents the value of that attribute defines the display name for that object that the PDM Workbench NX users can see.

File “**PWBSchema.xml**”:

```
...
<PWBSchema system="Aras" customization="Aras" displayName="NLS_System"
    visibleLength="15">
...

```

Configuration settings

Now the configuration of the PDM Workbench NX can be defined in the Schema file.

The tags are described in detail in the previous chapter. In this list you can see if the configuration is optional or mandatory.

xmap	optional	see “Exchange map” on page 21
soapTargetUrl	optional	see “SOAP target URL” on page 21
sessionSettings	optional	see “Session settings” on page 21
lastModificationDateAttribute	mandatory	see “Last modification date attribute” on page 22
createparts	optional	See “Create Part mode” on page 17

“object”: 1-n

This tag contains the definition of a PDM object class which can be used (queried, created, etc.) by the user.

The definition of PDM object classes, their corresponding dialogs and the actions that can be performed on them are described in the chapter **PDM Objects**.

“attribute”: 0-n

The definition of PDM attributes that are referenced in dialogs.

Attributes and dialog forms are explained in the chapter **PDM Attributes and Form Attributes**.

***“pwbAttribute”*: 0-n**

The definition of attributes that do not correspond directly to PDM attributes of PDM objects.

Attributes and dialog forms are explained in the chapter **PDM Attributes and Form Attributes**.

***“dataSource”*: 0-n**

Data sources contain attribute values. By assigning data sources to attributes default values for these attributes can be defined.

Data sources are explained in the chapter **Data Sources**.

PDM Attributes and Form Attributes

Every PDM attribute that is displayed in a dialog form should be defined in an *attribute* tag.

The *attribute* definition contains the following attributes:

- “name” Mandatory, must correspond to the PDM attribute's name.
- “displayName” Mandatory. As described in “NLS Support for Display Names” the NLS string for the “displayName” XML attribute.
- “dataSource” Optional. The data source includes the possible values for this attribute.
- “isFileName” Optional. If it is set to “true” the value of the corresponding input file name is checked about illegal² characters when creating a file.
- “isPartNumber” Optional.
- “autoName” Optional.
- “isDerived” Optional.

Example:

```
<attribute name="name" displayName="NLS_Name" isFileName="true"
isPartNumber="true" autoName="true"/>
<attribute name="current" displayName="NLS_current"
dataSource="LifeCycleStates"/>
<attribute name="revision" displayName="NLS_Revision" />
```

A form definition contains form attributes which reference the previously defined PDM attribute.

The *form* attribute definitions contain the following attributes:

- “name” Mandatory, must correspond to the PDM attribute's name.

² Filenames must not contain control characters, non printable characters and any of the following characters: *?;<|

- “displayName” Optional. If not defined here the display name of the *attribute* tag will be used.
- “mode” Possible values are “output” (read-only), “update” (can be modified), or “select” (e.g. for combo boxes).
Default is “output”.
- “visibleLength” Optional, the length of the text editor widget in characters.
- “allowedLength” Optional, the length of the value that can be inserted in the text editor widget in characters.
- “required” “true” or “false”. If “true”, then a value must be set.
Default is “false”.
- “widgetType” Possible values are “SingleLineEditor”, “MultiLineEditor”, “ComboBox”, “SingleCheckBox”, “CheckBoxes”, “RadioButtons”, “SingleSelectorList”, “MultiSelectorList”, “NameValueList”, “Date”.
Default is “SingleLineEditor”.
- “dataSource” Optional. The value defines the link to a data source that is more special than the linked data source in the <attribute> tag.
- “displayOnly” Possible values are “true” or “false”. If “true”, then the display value of the value of the data source will be used.

Example:

```
<form name="Query">
...
<formAttribute name="name" widgetType="SingleLineEditor"
mode="update" visibleLength="15" required="false" />
...
</form>
```

Form definitions generally refer to classes of PDM objects (query form, properties form, etc.). The definition of PDM object classes is described in chapter **PDM Objects**.

Description of the Widget Types

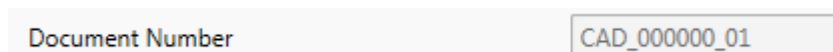
There are six different widget types available to build up dialogs with. All widget types except “SingleLineEditor”, “MultiLineEditor” and one mode of “NameValueList” can only be used on attributes that have certain kinds of Data Sources attached. Data Sources are a container of a limited set of values.

The detailed explanation of Data Sources you can find in chapter **Data Sources**.

SingleLineEditor Supports “update” and “output” mode.
Can be used for attributes with no data source attached and also for attributes with data sources of type “SingleValue”.



Picture 30: Single Line Editor Widget, update mode



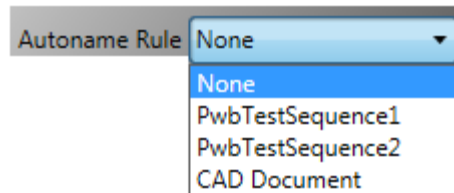
Picture 31: Single Line Editor Widget, output mode

MultiLineEditor Supports “update” and “output” mode.
Can be used for attributes with no data source attached and also for attributes with data sources of type “ValueList”.



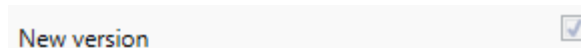
Picture 32: Multi Line Editor Widget, update mode

ComboBox Supports “select” and “output” mode.
This widget type can only be used for attributes with data sources of type “ValueList”, “BooleanValueList” or “invokeMessage” if this message returns a set of values.



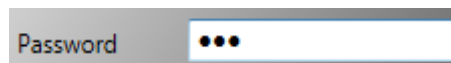
Picture 33: Combo Box Widget, select mode

SingleCheckBox Supports “select” and “output” mode.
Needs an attribute with a data source of type “BooleanValueList”.
This widget should be used only for required attributes or for attributes that are only displayed, already set to a value and cannot be updated.



Picture 34: Single Check Box Widget, select mode

PasswordBox Supports “update” mode.
Can be used for attributes with no data source.



Picture 35: Password Box Widget

URL Supports “output” mode.
Can be used for attributes with no data source.



Picture 36: URL Widget

Login Form

This tag contains a description of the “Login” form. It defines the attributes needed for logging in to the PDM system. Generally it contains the attributes “login name”, “password”, and “database” at least, though other attributes like “group” can be defined if it is necessary for the PDM system.

Example:

```

<form name="Login" info="ShowOnlyLoginData">
  <frame displayName="NLS_UserData">
    <pwbFormAttribute name="PWBServerURL" displayName="Server"
      widgetType="URL" mode="output"
      visibleLength="15" required="true" />
    <pwbFormAttribute name="PWLoginUser" displayName="Username"
      widgetType="SingleLineEditor"
      mode="update" visibleLength="15"
      required="true" entryAllowed="true" />
    <pwbFormAttribute name="PWLoginPassword"
      displayName="Password"
      widgetType="PasswordBox" mode="update"
      visibleLength="15" required="false" />
    <formAttribute name="LoginDatabase" displayName="Database"
      widgetType="ComboBox" mode="update"
      visibleLength="15" required="true"
      entryAllowed="false" />
    <formAttribute name="pwbAutonameRule"
      displayName="Autoname Rule"
      widgetType="ComboBox" mode="update"
      visibleLength="15" required="false"
      entryAllowed="false" />
  </frame>
</form>

```

The XML tags inside the *frame* tag describe how the attributes “user” and “password” are displayed in the login dialog.

Mandatory.

PDM Objects

XML tags *object* define PDM object classes that can be used in the PDM Workbench NX application. They represent the subset of objects defined within the PDM system which are needed in a PDM-CAD integration.

An *object* XML tag contains the following attributes:

- “name” The internal PDM class name.
- “displayName” The class name that is shown to the user.
- “icon” The icon that represents the class in the PDM window and the list window.

Example:

```
<object name="/Part/Assembly" displayName="NLS_Assembly" icon="Aras_Part">
```

Description of PDM Objects

The tag *description* defines which of the attributes of the class should be displayed beside the icon. In this example, these are the attributes “item_number”, “major_rev”, “generation”, “name”, and “state”.

Example:

```

<description>
  <descAttribute name="item_number" />
  <descAttribute name="major_rev" />
  <descAttribute name="generation" />

```

```

    <descAttribute name="name" />
    <descAttribute name="state" />
</description>

```

Actions on PDM Objects

Some actions that can be performed with PDM objects are defined in the Schema file.

Menu actions are defined with an *action* tag. The action “Query” can be defined on any object type.

Example:

```

<!-- * all PWB toolbar actions permitted for this object * -->
<action name="Query" />

```

If, for instance, the action “Query” is defined for the object type “/Part/Assembly”, then, when the user clicks on the “Query” menu button, the type “Assembly” (display name) is included in the query dialog list, otherwise it is not.

PDM Object Forms

The following forms can be defined for an object class:

“Query”, “Properties”

Example:

```

<form name="Query">
  <formAttribute name="item_number" displayName="Document Number"
    widgetType="SingleLineEditor" mode="update"
    visibleLength="15" required="false"
    listViewRelevant="true" entryAllowed="true"
    dataSource="RecentlyUsedValueDataSource" />
  <formAttribute name="major_rev" displayName="Major Rev."
    widgetType="SingleLineEditor" mode="update"
    visibleLength="15" required="false"
    listViewRelevant="true" entryAllowed="true"
    dataSource="RecentlyUsedValueDataSource" />
  <formAttribute name="generation" displayName="Generation"
    widgetType="SingleLineEditor" mode="update"
    visibleLength="15" required="false"
    listViewRelevant="true" entryAllowed="true"
    dataSource="RecentlyUsedValueDataSource" />
  <formAttribute name="name" displayName="Name"
    widgetType="SingleLineEditor" mode="update"
    visibleLength="15" required="false"
    listViewRelevant="true" entryAllowed="true"
    dataSource="RecentlyUsedValueDataSource" />
  <formAttribute name="state" displayName="State"
    widgetType="SingleLineEditor" mode="update"
    visibleLength="15" required="false"
    listViewRelevant="true" entryAllowed="true"
    dataSource="RecentlyUsedValueDataSource" />
  <formAttribute name="description" displayName="Description"
    widgetType="MultiLineEditor" mode="update"
    visibleLength="15" required="false"
    listViewRelevant="true" />
  <formAttribute name="created_on" displayName="Created on"
    widgetType="SingleLineEditor" mode="update"
    visibleLength="15" required="false"
    listViewRelevant="true" entryAllowed="true"
    dataSource="RecentlyUsedValueDataSource" />

```

```

<formAttribute name="modified_on" displayName="Modified on"
  widgetType="SingleLineEditor" mode="update"
  visibleLength="15" required="false"
  listViewRelevant="true" entryAllowed="true"
  dataSource="RecentlyUsedValueDataSource" />

<formAttribute name="created_by_id" displayName="Created by ID"
  widgetType="SingleLineEditor" mode="update"
  visibleLength="15" required="false"
  listViewRelevant="true" entryAllowed="true"
  dataSource="RecentlyUsedValueDataSource" />

<formAttribute name="modified_by_id"
  displayName="Modified by ID"
  widgetType="SingleLineEditor" mode="update"
  visibleLength="15" required="false"
  listViewRelevant="true" entryAllowed="true"
  dataSource="RecentlyUsedValueDataSource" />

<formAttribute name="locked_by_id" displayName="Locked by ID"
  widgetType="SingleLineEditor" mode="update"
  visibleLength="15" required="false"
  listViewRelevant="true" entryAllowed="true"
  dataSource="RecentlyUsedValueDataSource" />

</form>

```

Data Sources

Data sources describe a static set of values that are already known when writing the Schema file. The set of these values will never change during the lifetime of the PDM Workbench NX.

Data Source “Value” tag

The *value* tag of static data sources contains the following XML tags:

- “name” The PDM name of the attribute.
- “displayName” The dialog display name of the attribute.
- “booleanValue” “true” or “false” to assign the correct value to the attribute names (this tag is only used for type “BooleanValueList”).
- “valueName” The PDM name of the value attribute (this tag is only used for type “NameValueList”).
- “displayValue” The dialog display name of the value attribute (this tag is only used for type “NameValueList”).

Static data sources can be of type:

ValueList: the data source contains a set of static value elements.

Example:

```

<dataSource name="ItemTypes" type="ValueList">
  <value name="/CAD/Mechanical/Assembly" displayName="Assembly"/>
  <value name="/CAD/Mechanical/Part" displayName="Model" />
  <value name="/CAD/Mechanical/Drawing" displayName="Drawing" />
</dataSource>

```

BooleanValueList: the data source contains exactly the value pair “true” and “false”.

Example:

```

<dataSource name="TrueOrFalse" type="BooleanValueList">
  <value name="1" displayName="NLS_true" booleanValue="true" />
  <value name="0" displayName="NLS_false" booleanValue="false" />
</dataSource>

```

Complete example of using a data source tag:

The attribute “new_version” can be assigned exactly to “true” or “false”. Therefore we define a data source called “TrueOrFalse” and attach this container to the attribute description.

```

<attribute name="new_version" displayName="NLS_new_version"
  dataSource="TrueOrFalse" />

<dataSource name="TrueOrFalse" type="BooleanValueList">
  <value name="1" displayName="NLS_true" booleanValue="true" />
  <value name="0" displayName="NLS_false" booleanValue="false" />
</dataSource>

```

Customizing PDM Workbench NX Menu

The PDM Workbench NX menu is integrated by default into the standard NX menu dialog. But it is also possible to customize the NX menu to the individual needs.

It is possible to add PDM Workbench NX menu buttons to existing or new menus, toolbars or ribbon tabs. See the Manuscript User's Guide from the NX documentation.

The PDM Workbench NX also supports the replacement of the standard NX buttons by some PDM Workbench NX buttons.

The replacement is possible for the “Query”, “Update”, “Replace” and “Add Component” functions.

To do this, the “PWBSchema_Aras_NX.xml” and the “menu.men” file needs to be adapted.

Within the menu file the NX buttons are overwritten with the PDM Workbench NX action:

For example replace

```

BUTTON PDM_WORKBENCH_QUERY
LABEL Query
TOOLBAR_LABEL Query
MESSAGE Queries the PDM-System.
BITMAP PWB_Tlb_Query.bmp
SENSITIVITY OFF
ACTIONS BL_query

```

with

```

BUTTON UG_FILE_OPEN
LABEL Query
TOOLBAR_LABEL Query
MESSAGE Queries the PDM-System.
BITMAP PWB_Tlb_Query.bmp
SENSITIVITY OFF
ACTIONS BL_query

```

in the menu file.

In addition the ButtonName attribute of the "PWBSchema_Aras_NX.xml" configuration file needs to be adapted.

The displayName of the PDM Workbench NX function, here the 'Query' ButtonName, has to be replaced with the new button name (UG_FILE_OPEN).

```
<attribute name="ButtonName" displayName="Button" dataSource="ButtonNames"/>
<dataSource name="ButtonNames" type="ValueList">
  <value name="Query" displayName="UG_FILE_OPEN" />
  <value name="AddComponent" displayName="PDM_WORKBENCH_ADD_COMPONENT" />
  <value name="Replace" displayName="PDM_WORKBENCH_REPLACE" />
  <value name="update" displayName="PDM_WORKBENCH_UPDATE" />
  <value name="createFile" displayName="PDM_WORKBENCH_CREATEFILE" />
</dataSource>
```