



# TCI Release 3.3 Customizing Manual

---

## Copyright

© 2012, 2018 T-Systems International GmbH.  
All rights reserved. Printed in Germany.

---

## Issued by

T-Systems International GmbH  
GDC Product Lifecycle Management  
Fasanenweg 5  
70771 Leinfelden-Echterdingen  
Germany

---

## Contacts

web: <http://www.cmi-support.com>  
email: [cmi\\_support@t-systems.com](mailto:cmi_support@t-systems.com)  
phone: +49 (0) 40 30600-5544

---

## Brief details

Customizing manual for TCI.

This guide describes the customizing of TCI.

Before using this guide, be sure you understand:

- the administration of 3DEXPERIENCE,
  - the administration of Teamcenter,
  - the administration of TCI,
  - the Java programming language.
- 

## Trademarks

3DEXPERIENCE is a registered trademark of Dassault Systèmes.

Teamcenter is a registered trademark of Siemens PLM Corporation.

Oracle is a registered trademark of Oracle Corporation.

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

---

# Table of Contents

- 1 Overview ..... 1**
- 1.1 Introduction Customizing TCI ..... 1
- 1.2 Customization Tasks ..... 1
- 2 Customization Options ..... 3**
- 2.1 Customizing change location of control (CLOC) ..... 3
- 2.1.1 CLOC to 3DEXPERIENCE ..... 3
- 2.1.3 CLOC to XPDM ..... 4
- 2.2 Customizing attribute mapping ..... 5
- 2.3 Customizing attribute value mapping ..... 11
- 2.4 Customizing dataset filter ..... 14
- 2.5 Customizing dataset naming ..... 15
- 2.6 Customizing item type ..... 18
- 2.7 Customizing lifecycle and maturity mapping ..... 20
- 2.8 Customizing part number mapping ..... 25
- 2.9 Customizing revision mapping ..... 26
- 2.10 Customizing type mapping ..... 29
- 2.11 Customizing user mapping ..... 33
- 2.12 Customizing item reuse ..... 37
- 2.13 Customizing PDM import ..... 38
- 2.14 Customizing 3DEXPERIENCE property info ..... 39
- 3 Customizing Example ..... 40**

## List of Tables

Table 1: Change the location of control to 3DEXPERIENCE .....	3
Table 2: Finish change the location of control to 3DEXPERIENCE.....	3
Table 3: Change the location of control to XPDM .....	4
Table 4: Finish change the location of control to XPDM.....	4
Table 5: Standard attribute value Product ID for 3DEXPERIENCE.....	5
Table 6: Standard attribute value Product Name for 3DEXPERIENCE.....	5
Table 7: Standard attribute value Product Description for 3DEXPERIENCE.....	6
Table 8: Standard attribute value Drawing ID for 3DEXPERIENCE.....	6
Table 9: Standard attribute value Drawing Name for 3DEXPERIENCE.....	7
Table 10: Standard attribute value Drawing Description for 3DEXPERIENCE.....	7
Table 11: Standard attribute value Rep3D ID for 3DEXPERIENCE.....	8
Table 12: Standard attribute value Rep3D Name for 3DEXPERIENCE.....	8
Table 13: Standard attribute value Rep3D Description for 3DEXPERIENCE.....	9
Table 14: Standard attribute value Document ID for 3DEXPERIENCE.....	9
Table 15: Standard attribute value Document Description for 3DEXPERIENCE.....	10
Table 16: Attribute value for PDM import.....	11
Table 17: Attribute value for 3DEXPERIENCE import .....	12
Table 18: Converted attribute values for 3DEXPERIENCE import.....	13
Table 19: Attribute list indication.....	13
Table 20: Dataset filtering for the 3DEXPERIENCE import.....	14
Table 21: Dataset naming for the 3DEXPERIENCE import .....	15
Table 22: Dataset naming for the PDM import.....	16
Table 23: Dataset file naming for the PDM import .....	17
Table 24: Item type for creating items in PDM .....	18
Table 25: Reuse Item type for creating items in PDM.....	18
Table 26: Item type for creating Fastener Items in PDM.....	19
Table 27: 3DEXPERIENCE lifecycle definition .....	20
Table 28: 3DEXPERIENCE maturity definition .....	20
Table 29: 3DEXPERIENCE document lifecycle definition.....	21
Table 30: 3DEXPERIENCE document maturity definition.....	22
Table 31: 3DEXPERIENCE generic document lifecycle definition .....	22
Table 32: 3DEXPERIENCE generic document maturity definition .....	23
Table 33: PDM value for object immutable .....	23
Table 34: PDM status list for a given 3DEXPERIENCE maturity .....	24
Table 35: 3DEXPERIENCE Part Number mapping .....	25
Table 36: PDM Part Number mapping.....	25

Table 37: 3DEXPERIENCE Revision Index.....	26
Table 38: PDM Revision ID .....	27
Table 39: Use of unmapped Item Revision in PDM .....	28
Table 40: Retrieve XPDMXML mapping type for an Item Revision .....	29
Table 41: Retrieve XPDMXML mapping type for an Item Revision .....	30
Table 42: Retrieve XPDMXML mapping type for an Item Revision Instance.....	30
Table 43: Retrieve XPDMXML mapping type for a Dataset .....	31
Table 44: Retrieve XPDMXML mapping type for a Dataset .....	31
Table 45: Retrieve XPDMXML mapping type for a Dataset Instance.....	32
Table 46: Retrieve XPDMXML mapping type for a Dataset Instance.....	32
Table 47: Retrieve value for Owner for 3DEXPERIENCE import.....	33
Table 48: Retrieve value for Organization for 3DEXPERIENCE import .....	34
Table 49: Retrieve value for Project for 3DEXPERIENCE import .....	34
Table 50: Retrieve the Owning User for PDM import .....	35
Table 51: Retrieve the Owning Group for PDM import.....	35
Table 52: Retrieve list of additional objects for change owner function for PDM import .....	36
Table 53: Creation of a new reusable Item Revision in PDM .....	37
Table 54: Reading of a reusable Item Revision from PDM .....	37
Table 55: Prepare newly created PDM Item for the import .....	38
Table 56: Prepare newly created PDM Item Revision for the import.....	38
Table 57: Define the attributes retrieved from 3DEXPERIENCE for the Show 3DEXPERIENCE dialog in PDM.....	39

# 1 Overview

## 1.1 Introduction Customizing TCI

This document describes the technique required to customize TCI.

To perform the customizing, you should be familiar with:

- TCI,
- Java programming language,
- Java development tool (e.g. Eclipse),
- creation and deployment of JAR files.

The current version of TCI covers the following customizing topics:

- change location of control,
- attribute mapping,
- attribute value mapping,
- dataset filter,
- dataset naming,
- item types,
- maturity mapping,
- part number mapping,
- revision mapping,
- type mapping,
- user mapping,
- item reuse,
- PDM import,
- 3DEXPERIENCE property info.

If you need other customizing for your needs, don't hesitate to contact our support [cmi\\_support@t-systems.com](mailto:cmi_support@t-systems.com).

## 1.2 Customization Tasks

For customizing TCI the following steps have to be performed:

- create Java overwrite method(s),
- create your custom JAR file,
- deploy your custom JAR file into the web application server,
- register your customization in the TCI setting.

An example about customizing can be found in chapter *Customizing Example*. Further information about registration can be found in chapter *Customization registration* in the TCI installation manual.

## 2 Customization Options

### 2.1 Customizing change location of control (CLOC)

#### 2.1.1 CLOC to 3DEXPERIENCE

To adapt the behavior of changing the location of control (CLOC) to 3DEXPERIENCE, implement one or more of the following extension points.

---

<b>Description:</b>	The interface function is a part of the service CLOC from XPDM to 3DEXPERIENCE and prepares the CLOC.
<b>Function:</b>	void <b>changeLOCtoV6</b> ( String... obids )
<b>Class:</b>	ChangeLocationOfControl
<b>Package:</b>	com.tsystems.tci.customization
<b>Parameter:</b>	String... obids the Object IDs to change LOC
<b>Default:</b>	Default implementation performs no action.

---

Table 1: Change the location of control to 3DEXPERIENCE

---

<b>Description:</b>	The interface function is a part of the service CLOC from XPDM to 3DEXPERIENCE and finishes the CLOC.
<b>Function:</b>	void <b>finishChangeLOCtoV6</b> ( String... obids )
<b>Class:</b>	ChangeLocationOfControl
<b>Package:</b>	com.tsystems.tci.customization
<b>Parameter:</b>	String... obids the Object IDs to finish the change LOC
<b>Default:</b>	In the default implementation the ownership of all Object IDs will be transferred to 3DEXPERIENCE in the TCI database.
<b>Notes:</b>	See function <b>changeLOCtoV6</b> .

---

Table 2: Finish change the location of control to 3DEXPERIENCE



### 2.1.3 CLOC to XPDM

To adapt the behavior of changing the location of control (CLOC) to XPDM, implement one or more of the following extension points.

---

<b>Description:</b>	The interface function is a part of the service CLOC from 3DEXPERIENCE to XPDM and prepares the CLOC.
<b>Function:</b>	void <b>changeLOCtoXPDM</b> ( String... obids )
<b>Class:</b>	ChangeLocationOfControl
<b>Package:</b>	com.tsystems.tci.customization
<b>Parameter:</b>	String... obids the Object IDs to change LOC
<b>Default:</b>	Default implementation performs no action.

---

Table 3: Change the location of control to XPDM

---

<b>Description:</b>	The interface function is a part of the service CLOC from 3DEXPERIENCE to XPDM and finishes the CLOC.
<b>Function:</b>	void <b>finishChangeLOCtoXPDM</b> ( String... obids )
<b>Class:</b>	ChangeLocationOfControl
<b>Package:</b>	com.tsystems.tci.customization
<b>Parameter:</b>	String... obids the Object IDs to finish the change LOC
<b>Default:</b>	In the default implementation the ownership of all Object IDs will be transferred to XPDM in the TCI database.
<b>Notes:</b>	See function <b>changeLOCtoXPDM</b> .

---

Table 4: Finish change the location of control to XPDM







To adapt the Rep3D ID mapping, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve the standard attribute value for the Rep3D ID from PDM.		
<b>Function:</b>	String <b>getIDForV6</b> ( com.tsystems.tci.xpdmxml.Rep3D rep3d, String defaultString )		
<b>Class:</b>	CustomAttributeMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	Rep3D	rep3d	the Rep3D object
	String	defaultString	the default string
	return		the ID to be set for the Rep3D ID
<b>Default:</b>	The default implementation returns the value of the input parameter defaultString.		

Table 11: Standard attribute value Rep3D ID for 3DEXPERIENCE

To adapt the Rep3D Name mapping, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve the standard attribute value for the Rep3D Name from PDM.		
<b>Function:</b>	String <b>getNameForV6</b> ( com.tsystems.tci.xpdmxml.Rep3D rep3d, String defaultString )		
<b>Class:</b>	CustomAttributeMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	Rep3D	rep3d	the Rep3D object
	String	defaultString	the default string
	return		the Name to be set for Rep3D Name
<b>Default:</b>	The default implementation returns the value of the input parameter defaultString.		

Table 12: Standard attribute value Rep3D Name for 3DEXPERIENCE



To adapt the Document Description mapping, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve the standard attribute value for the Document Description from PDM.		
<b>Function:</b>	String <b>getDescriptionForV6</b> ( com.tsystems.tci.xpdmxml.XpdmDocumentObject doc, String defaultString )		
<b>Class:</b>	CustomAttributeMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	XpdmDocumentObject	doc	the object
	String	defaultString	the default string
	return		the Description to be set for the Document Description
<b>Default:</b>	The default implementation returns the value of the input parameter defaultString.		

Table 15: Standard attribute value Document Description for 3DEXPERIENCE

## 2.3 Customizing attribute value mapping

To adapt the attribute value mapping for the PDM import, implement the following extension point. To be able to change the attribute values, the attributes must be defined as custom attributes.

For further information about custom attributes, please refer to TCI installation manual.

---

<b>Description:</b>	The interface function is called to retrieve an attribute value for PDM import. This function is called each time a custom attribute is transferred to PDM.		
---------------------	---	--	--

---

<b>Function:</b>	<pre>String[] <b>getImportValue</b> (     final com.tsystems.tci.pdm.interfaces.IPdmSession      tcs,     final com.tsystems.tci.pdm.interfaces.IPdmModelObject tcObject,     final String      xName,     final String      tcName,     final String      type,     final String      xmlValue )</pre>		
------------------	---	--	--

---

<b>Class:</b>	CustomAttributeValue		
---------------	----------------------	--	--

---

<b>Package:</b>	com.tsystems.tci.customization		
-----------------	--------------------------------	--	--

---

<b>Parameter:</b>	IPdmSession	tcs	the PDM Session to use
	IPdmModelObject	tcObject	the PDM object
	String	xName	the 3DEXPERIENCE property name
	String	tcName	the PDM property name
	String	type	the type
	String	xmlValue	the xml property value
	return		the xml property values

---

<b>Default:</b>	The default implementation returns the value of the input parameter xmlValue.		
-----------------	---	--	--

---

Table 16: Attribute value for PDM import



To adapt the attribute value mapping for the 3DEXPERIENCE import, implement the following extension point. To be able to change the attribute values, the attributes must be defined as custom attributes.

For further information about custom attributes, please refer to TCI installation manual.

---

**Description:** The interface function is called to retrieve an attribute value for 3DEXPERIENCE import. This function is called each time a custom attribute is transferred to 3DEXPERIENCE.

---

**Function:** String[] **getExportValue** (  
     final com.tsystems.tci.pdm.interfaces.IPdmSession        tcs,  
     final com.tsystems.tci.pdm.interfaces.IPdmModelObject tcObject,  
     final String    xName,  
     final String    tcName,  
     final String    type )

---

**Class:** CustomAttributeValue

---

**Package:** com.tsystems.tci.customization

---

<b>Parameter:</b>	IPdmSession	tcs	the PDM Session to use
	IPdmModelObject	tcObject	the PDM object
	String	xName	the 3DEXPERIENCE property name
	String	tcName	the PDM property name
	String	type	the type
	return		the xml property values

---

**Default:** Returns the current values from PDM.

---

Table 17: Attribute value for 3DEXPERIENCE import

To adapt the converting of attribute values for the 3DEXPERIENCE import, implement the following extension point:

<b>Description:</b>	Get the converted attribute value for 3DEXPERIENCE import. This function is called each time a custom attribute is transferred to 3DEXPERIENCE.		
<b>Function:</b>	String[] <b>getConvertedExportValue</b> ( String        xName, String        tcName, String        type, String[]      values )		
<b>Class:</b>	CustomAttributeValue		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	String	xName	3DEXPERIENCE property name
	String	tcName	the PDM property name
	String	type	the type
	String[]	values	the xml property values
	return		the xml property values
<b>Default:</b>	The default implementation returns the value of the input parameter values.		

Table 18: Converted attribute values for 3DEXPERIENCE import

To adapt the attribute list indication, implement the following extension point:

<b>Description:</b>	Ending of list types. This function is called each time a custom attribute is transferred. Return value indicates if the given type is a list.		
<b>Function:</b>	boolean <b>isListType</b> ( String type )		
<b>Class:</b>	CustomAttributeValue		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Description:</b>	String	type	to check
	return		true, if type is a list type
<b>Default:</b>	True, if parameter <i>type</i> ends with substring "list"		

Table 19: Attribute list indication

## 2.4 Customizing dataset filter

To adapt the dataset filtering for the 3DEXPERIENCE import, implement the following extension point:

<b>Description:</b>	Customizing the dataset filtering. This function is called each time a dataset is transferred to 3DEXPERIENCE.		
<b>Function:</b>	<pre>com.tsystems.tci.pdm.interfaces.IPdmModelObject [] filterDatasets (     com.tsystems.tci.pdm.interfaces.IPdmSession          session     com.tsystems.tci.pdm.interfaces.IPdmModelObject      revision,     com.tsystems.tci.pdm.interfaces.IPdmModelObject []  relatedDatasets )</pre>		
<b>Class:</b>	CustomDatasetFilter		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	IPdmSession	session	the Session
	IPdmModelObject	revision	the Revision
	IPdmModelObject []	relatedDatasets	the expanded datasets
	return		the filtered datasets
<b>Default:</b>	No filtering.		

Table 20: Dataset filtering for the 3DEXPERIENCE import

## 2.5 Customizing dataset naming

To adapt dataset naming for the 3DEXPERIENCE import, implement the following extension point:

<b>Description:</b>	Customizing the dataset naming. This function is called each time a dataset is transferred to 3DEXPERIENCE.		
<b>Function:</b>	<pre>void updateDatasetNames (     com.tsystems.tci.db.tciDB1.DBAdapter dbAdapter,     String partNumber,     String revision,     List&lt;String&gt; dsObids,     List&lt;String&gt; dsNames,     List&lt;String&gt; dsTypes,     List&lt;String&gt; fileNames,     List&lt;String&gt; extensions,     List&lt;String&gt; repNames,     List&lt;Boolean&gt; isPart3Ds )</pre>		
<b>Class:</b>	CustomDatasetNaming		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	DBAdapter	dbAdapter	the context object
	String	partNumber	the Part Number
	String	revision	the Revision
	List<String>	dsObids	the dataset obids
	List<String>	dsNames	the dataset names
	List<String>	dsTypes	the dataset types
	List<String>	fileNames	the dataset file names
	List<String>	extensions	the dataset extensions
	List<String>	repNames	the dataset representation names
	List<Boolean>	isPart3Ds	the dataset is 3DPart flag
<b>Default:</b>	The default implementation sets the Part3D to a unique Name for all revisions, and renames all duplicate dataset names.		

Table 21: Dataset naming for the 3DEXPERIENCE import



To adapt the dataset file naming for the PDM import, implement the following extension point:

<b>Description:</b>	Customizing dataset file naming. The interface function is called to get the dataset file name for the PDM import.		
<b>Function:</b>	String <b>getTCFileName</b> ( Dataset                  datasetDb, XpdmFileObject      fileObject, XpdmXml              xpdmxml, List<String>          fileNames )		
<b>Class:</b>	CustomDatasetNaming		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	Dataset	datasetDb	the dataset TCI DB
	XpdmFileObject	fileObject	the xpdm file object
	XpdmXml	xpdmxml	the xpdmxml object
	List<String>	fileNames	the used file names
<b>Default:</b>	The default implementation returns partnumber_revIndex.extension		

Table 23: Dataset file naming for the PDM import

## 2.6 Customizing item type

To adapt the item type for creating Items in PDM, implement the following extension point:

---

<b>Description:</b>	The interface function is called to retrieve the Item Type for Item creation in PDM.		
<b>Function:</b>	String <b>getItemTypeForCreate</b> ( com.tsystems.tci.xpdmxml.Product prod )		
<b>Class:</b>	CustomItemType		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	Product	prod	the product to get Item type for return the Item type
<b>Default:</b>	The default implementation returns “Item” as item type for creating new items in PDM.		

---

Table 24: Item type for creating items in PDM

To adapt the reuse item type for creating items in PDM, implement the following extension point:

---

<b>Description:</b>	The interface function is called to retrieve the reuse Item Type for creating Items in PDM.		
<b>Function:</b>	String <b>getReuseItemTypeForCreate</b> ()		
<b>Class:</b>	CustomItemType		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	return		the Item Type
<b>Default:</b>	The default implementation returns “Item” as item type for creating reuse items in PDM.		

---

Table 25: Reuse Item type for creating items in PDM

To adapt the item type for creating Fastener Items in PDM, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve the item type for Fastener Item creation in PDM.		
<b>Function:</b>	String <b>getItemTypeForFastener</b> ( com.tsystems.tci.xpdmxml.Product prod )		
<b>Class:</b>	CustomItemType		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	Product	prod	the Product to get Item Type for return the Item Type
<b>Default:</b>	The default implementation returns "WeldPoint" as Item Type for creating new Fastener Items in PDM.		

Table 26: Item type for creating Fastener Items in PDM



## 2.7 Customizing lifecycle and maturity mapping

To adapt the value for 3DEXPERIENCE Lifecycle, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve the value for Product Lifecycle and Fastener Product Lifecycle from PDM. The value is used for import in 3DEXPERIENCE.		
<b>Function:</b>	String <b>getLifecycle</b> ( String project, String releaseStates )		
<b>Class:</b>	CustomMaturityMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	String	project	the Project Name in PDM
	String	releaseStates	the Release States in PDM
	return		the value for 3DEXPERIENCE Lifecycle
<b>Default:</b>	The default implementation returns the defined value from TCI setting file <i>TCISettings.xml</i> . If no TCI setting is defined, the value "VPLM_SMB" is returned.  For further information about the setting file, please refer to the chapter <i>TCI configuration</i> in the TCI installation manual.		

Table 27: 3DEXPERIENCE lifecycle definition

To adapt the value for 3DEXPERIENCE Maturity, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve the value for Product Maturity and Fastener Product Maturity from PDM. The value is used for import in 3DEXPERIENCE.		
<b>Function:</b>	String <b>getMaturity</b> ( String project, String releaseStates )		
<b>Class:</b>	CustomMaturityMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	String	project	the Project Name in PDM
	String	releaseStates	the Release States in PDM
	return		the value for 3DEXPERIENCE Maturity
<b>Default:</b>	The default implementation returns the defined value from TCI setting file <i>TCISettings.xml</i> . If no TCI setting is defined, the value "SHARED" is returned.  For further information about the setting file, please refer to the chapter <i>TCI configuration</i> in the TCI installation manual.		

Table 28: 3DEXPERIENCE maturity definition

To adapt the value for 3DEXPERIENCE Document Lifecycle, implement the following extension point:

---

**Description:** The interface function is called to retrieve the value for Document Lifecycle from PDM. The value is used for import in 3DEXPERIENCE.

---

**Function:** String **getDocumentLifecycle** (  
String project,  
String releaseStates )

---

**Class:** CustomMaturityMapping

---

**Package:** com.tsystems.tci.customization

---

<b>Parameter:</b>	String	project	the Project Name in PDM
	String	releaseStates	the Release States in PDM
	return		the value for 3DEXPERIENCE Document Lifecycle

---

**Default:** The default implementation returns the defined value from TCI setting file *TCISettings.xml*. If no TCI setting is defined, the value "Document" is returned.

For further information about the setting file, please refer to the chapter *TCI configuration* in the TCI installation manual.

---

Table 29: 3DEXPERIENCE document lifecycle definition

To adapt the value for 3DEXPERIENCE Document Maturity, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve the value for Document Maturity from PDM. The value is used for import in 3DEXPERIENCE.		
<b>Function:</b>	String <b>getDocumentMaturity</b> ( String project, String releaseStates )		
<b>Class:</b>	CustomMaturityMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	String	project	the Project Name in PDM
	String	releaseStates	the Release States in PDM
	return		the value for 3DEXPERIENCE Document Maturity
<b>Default:</b>	The default implementation returns the defined value from TCI setting file <i>TCISettings.xml</i> . If no TCI setting is defined, the value "Exists" is returned. For further information about the setting file, please refer to the chapter <i>TCI configuration</i> in the TCI installation manual.		

Table 30: 3DEXPERIENCE document maturity definition

To adapt the value for 3DEXPERIENCE Generic Document Lifecycle, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve the value for Generic Document Lifecycle from PDM. The value is used for import in 3DEXPERIENCE.		
<b>Function:</b>	String <b>getGenericDocumentLifecycle</b> ( String project, String releaseStates )		
<b>Class:</b>	CustomMaturityMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	String	project	the Project Name in PDM
	String	releaseStates	the Release States in PDM
	return		the value for 3DEXPERIENCE Generic Document Lifecycle
<b>Default:</b>	The default implementation returns the defined value from TCI setting file <i>TCISettings.xml</i> . If no TCI setting is defined, the value "Controlled Production Release Rev2" is returned. For further information about the setting file, please refer to the chapter <i>TCI configuration</i> in the TCI installation manual.		

Table 31: 3DEXPERIENCE generic document lifecycle definition

To adapt the value for 3DEXPERIENCE Generic Document Maturity, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve the value for Generic Document Maturity from PDM. The value is used for import in 3DEXPERIENCE.		
<b>Function:</b>	String <b>getGenericDocumentMaturity</b> ( String project, String releaseStates )		
<b>Class:</b>	CustomMaturityMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	String	project	the Project Name in PDM
	String	releaseStates	the Release States in PDM
	return		the value for 3DEXPERIENCE Generic Document Maturity
<b>Default:</b>	The default implementation returns the defined value from TCI setting file <i>TCISettings.xml</i> . If no TCI setting is defined, the value "WIP" is returned. For further information about the setting file, please refer to the chapter <i>TCI configuration</i> in the TCI installation manual.		

Table 32: 3DEXPERIENCE generic document maturity definition

To adapt the value indicating if a specific PDM object is immutable, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve the value indicating if a specific status is immutable.		
<b>Function:</b>	String <b>isStatusImmutable</b> ( String project, String releaseStates )		
<b>Class:</b>	CustomMaturityMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	String	project	the Project name in PDM
	String	releaseStates	the Release states in PDM
	return		"TRUE", if the Item Revision has a status
<b>Default:</b>	The default implementation returns TRUE, if parameter releaseStates contains a status value.		

Table 33: PDM value for object immutable

To adapt the PDM status list for a given 3DEXPERIENCE Maturity, implement the following extension point:

---

<b>Description:</b>	The interface function is called to retrieve the PDM status list for a given 3DEXPERIENCE Maturity. This function is called each time a Product, Rep3D, Drawing, Document and Generic Document is created or updated in 3DEXPERIENCE and Maturity value is defined.		
<b>Function:</b>	String <b>getPDMStatusForV6Maturity</b> ( String maturity )		
<b>Class:</b>	CustomMaturityMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	String return	maturity	the 3DEXPERIENCE Maturity the PDM Status names or null if no status is configured
<b>Default:</b>	The return value is null for no status or the current PDM status list for a given V6 Maturity.		

---

Table 34: PDM status list for a given 3DEXPERIENCE maturity

## 2.8 Customizing part number mapping

To adapt the 3DEXPERIENCE Part Number, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve the Part Number for 3DEXPERIENCE.		
<b>Function:</b>	String <b>getPartNumberFor3DEXPERIENCE</b> ( String        type, String        defaultId )		
<b>Class:</b>	CustomPartNumberMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	String	type	the PDM Type
	String	defaultId	the PDM Part Number
	return		the new Part Number for 3DEXPERIENCE
<b>Default:</b>	The default implementation returns the Part Number based on the PartNumberMapping setting.		

Table 35: 3DEXPERIENCE Part Number mapping

To adapt the PDM Part Number, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve the Part Number for PDM.		
<b>Function:</b>	String <b>getPartNumberForPDM</b> ( String        type, String        defaultId )		
<b>Class:</b>	CustomPartNumberMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	String	type	the 3DEXPERIENCE Type
	String	defaultId	the 3DEXPERIENCE Part Number
	return		the new Part Number for PDM
<b>Default:</b>	The default implementation returns the Part Number based on the PartNumberMapping setting.		

Table 36: PDM Part Number mapping

## 2.9 Customizing revision mapping

To adapt the value for 3DEXPERIENCE Revision Index, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve the value for a 3DEXPERIENCE Revision.		
<b>Function:</b>	<pre>int <b>getRevisionIndex</b> (     final List&lt;Integer&gt; existingIndexList,     final String revisionString )</pre>		
<b>Class:</b>	CustomRevisionMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	final List<Integer>	existingIndexList	known indices from the TCI DB
	final String	revisionString	the Item Revision string from PDM
	return		the value for 3DEXPERIENCE Revision Index
<b>Default:</b>	The default implementation returns the next free index.		

Table 37: 3DEXPERIENCE Revision Index

To adapt the value for PDM Revision Index, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve the value for a PDM Revision ID.		
<b>Function:</b>	String <b>getRevisionID</b> ( final com.tsystems.tci.pdm.interfaces.IPdmSession           session, final com.tsystems.tci.xpdmxml.Product                    product, final com.tsystems.tci.pdm.interfaces.IPdmModelObject   item, final String    itemType )		
<b>Class:</b>	CustomRevisionMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	final IPdmSession	session	the PDM Session to use
	final Product	product	the Product to get PDM Revision for
	final IPdmModelObject	item	The Item to get Revision for. May be <i>null</i> when Item is not existing yet.
	final String	itemType	the Item Type of the Item
	return		the new Revision ID for PDM
<b>Default:</b>	The default implementation returns a new Revision ID.		

Table 38: PDM Revision ID



To adapt the value for use of unmapped Item Revision, implement the following extension point:

<b>Description:</b>	The interface function is called to decide if an Item Revision without mapping in the TCI database should be used for overwrites.		
<b>Function:</b>	<pre>boolean useUnmappedItemRev (     final com.tsystems.tci.pdm.interfaces.IPdmSession    session,     final com.tsystems.tci.xpdmxml.Product              product,     final com.tsystems.tci.pdm.interfaces.IPdmModelObject itemRev )</pre>		
<b>Class:</b>	CustomRevisionMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	final IPdmSession	session	the PDM Session to use
	final Product	product	the Product to import
	final IPdmModelObject	itemRev	the Item Revision to overwrite
	return		<i>true</i> , if the Revision can be used for overwrite
<b>Default:</b>	The default implementation returns <i>true</i> , if the Item Revision has no status and the Item Revision user and group match the current session user and the current session group. In all other cases, <i>false</i> is returned.		

Table 39: Use of unmapped Item Revision in PDM



<b>Description:</b>	The interface function is called to retrieve the XPDMXML mapping type for a specific PDM object class.		
<b>Function:</b>	String <b>getItemRevisionMappingType</b> ( String tcltemType, String tcltemID, String tcOBID )		
<b>Class:</b>	CustomTypeMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	String	tcltemType	the PDM Item Revision class name
	String	tcltemID	the PDM Item ID
	String	tcOBID	the PDM Item Revision OBID
	return		the XPDMXML mapping type string
<b>Default:</b>	The default implementation returns the defined value from TCI setting file <i>TCISettings.xml</i> . If no TCI setting is defined, the value "DefaultCustomization" is returned.  For further information about the setting file, please refer to the chapter <i>TCI Configuration</i> in the TCI installation manual.		

Table 41: Retrieve XPDMXML mapping type for an Item Revision

<b>Description:</b>	The interface function is called to retrieve the XPDMXML mapping type for a PDM Item Revision Instance.		
<b>Function:</b>	String <b>getItemRevisionInstMappingType</b> ( com.tsystems.tci.xpdmxml.Product parentProduct, com.tsystems.tci.xpdmxml.Product childProduct )		
<b>Class:</b>	CustomTypeMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	Product	parentProduct	the top Product to get mapping type between
	Product	childProduct	the bottom Product to get mapping type between
	return		the XPDMXML mapping type string
<b>Default:</b>	The default implementation returns the defined value from TCI setting file <i>TCISettings.xml</i> . If no TCI setting is defined, the value "DefaultCustomization" is returned.  For further information about the setting file, please refer to the chapter <i>TCI Configuration</i> in the TCI installation manual.		

Table 42: Retrieve XPDMXML mapping type for an Item Revision Instance

<b>Description:</b>	The interface function is called to retrieve the XPDMXML mapping type for a PDM Dataset.		
<b>Function:</b>	final String <b>getDatasetMappingType</b> ( com.tsystems.tci.pdm.interfaces.IPdmModelObject      dataset, com.tsystems.tci.pdm.interfaces.IPdmModelObject      parentItemRev)		
<b>Class:</b>	CustomTypeMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	IPdmModelObject	dataset	the dataset to get mapping type for
	IPdmModelObject	parentItemRev	the Item Revision to get mapping type for
	return		the XPDMXML mapping type
<b>Default:</b>	The default implementation calls the getDatasetMappingType (String datasetClassname, String itemRevisionClassname) function which returns the defined value from TCI setting file <i>TCISettings.xml</i> . If no TCI setting is defined, the value "DefaultCustomization" is returned.  For further information about the setting file, please refer to the chapter <i>TCI Configuration</i> in the TCI installation manual.		

Table 43: Retrieve XPDMXML mapping type for a Dataset

<b>Description:</b>	The interface function is called to retrieve the XPDMXML mapping type for a PDM Dataset.		
<b>Function:</b>	String <b>getDatasetMappingType</b> ( String datasetType, String itemRevisionType )		
<b>Class:</b>	CustomTypeMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	String	datasetType	the dataset type
	String	itemRevisionType	the Item Revision type
	return		the XPDMXML mapping type
<b>Default:</b>	The default implementation returns the defined value from TCI setting file <i>TCISettings.xml</i> . If no TCI setting is defined, the value "DefaultCustomization" is returned.  For further information about the setting file, please refer to the chapter <i>TCI Configuration</i> in the TCI installation manual.		

Table 44: Retrieve XPDMXML mapping type for a Dataset

<b>Description:</b>	The interface function is called to retrieve the XPDMXML mapping type for a PDM Dataset Instance.		
<b>Function:</b>	String <b>getDatasetInstMappingType</b> ( com.tsystems.tci.pdm.interfaces.IPdmModelObject     parentItemRev, com.tsystems.tci.pdm.interfaces.IPdmModelObject     dataset )		
<b>Class:</b>	CustomTypeMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	IPdmModelObject	parentItemRev	the parent Item Revision object
	IPdmModelObject	dataset	the dataset object
	return		the XPDMXML mapping type
<b>Default:</b>	The default implementation calls the getDatasetInstMappingType (String parentItemRevisionType, String datasetType) function which returns the defined value from TCI setting file <i>TCISettings.xml</i> . If no TCI setting is defined, the value "DefaultCustomization" is returned.  For further information about the setting file, please refer to the chapter <i>TCI Configuration</i> in the TCI installation manual.		

Table 45: Retrieve XPDMXML mapping type for a Dataset Instance

<b>Description:</b>	The interface function is called to retrieve the XPDMXML mapping type for a PDM Dataset Instance.		
<b>Function:</b>	String <b>getDatasetInstMappingType</b> ( String           parentItemRevisionType, String           datasetType )		
<b>Class:</b>	CustomTypeMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	String	parentItemRevisionType	the parent Item Revision type
	String	datasetType	the dataset type
	return		the XPDMXML mapping type
<b>Default:</b>	The default implementation returns the defined value from TCI setting file <i>TCISettings.xml</i> . If no TCI setting is defined, the value "DefaultCustomization" is returned.  For further information about the setting file, please refer to the chapter <i>TCI Configuration</i> in the TCI installation manual.		

Table 46: Retrieve XPDMXML mapping type for a Dataset Instance

## 2.11 Customizing user mapping

To adapt the value to set for *XPDMXML Owner* for 3DEXPERIENCE data import, implement the following extension point:

---

<b>Description:</b>	The interface function is called to retrieve the value to set for <i>XPDMXML Owner</i> for 3DEXPERIENCE data import from PDM Project.		
<b>Function:</b>	String <b>getOwnerForV6</b> ( String        project, String        owner )		
<b>Class:</b>	CustomUserMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	String	project	the PDM Project
	String	owner	the PDM Owner
	return		the XPDMXML Owner to set
<b>Default:</b>	The default implementation returns the Owner of the PDM project defined in TCI setting file <i>TCISettings.xml</i> . If no TCI setting is defined, an empty string is returned.  For further information about the TCI setting file, please refer to the chapter <i>TCI Configuration</i> in the TCI installation manual.		

---

Table 47: Retrieve value for Owner for 3DEXPERIENCE import

To adapt the value to set for *XPDMXML Organization* for 3DEXPERIENCE data import, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve the value to set for <i>XPDMXML Organization</i> for 3DEXPERIENCE data import from PDM Project.		
<b>Function:</b>	String <b>getOrganizationForV6</b> ( String          project, String          owner )		
<b>Class:</b>	CustomUserMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	String	project	the PDM Project
	String	owner	the PDM Owner
	return		the XPDMXML Organization to set
<b>Default:</b>	The default implementation returns the Organization of the PDM project defined in TCI setting file <i>TCISettings.xml</i> . If no TCI setting is defined, an empty string is returned.  For further information about the TCI setting file, please refer to the chapter <i>TCI Configuration</i> in the TCI installation manual.		

Table 48: Retrieve value for Organization for 3DEXPERIENCE import

To adapt the value to set for *XPDMXML Project* for 3DEXPERIENCE data import, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve the value to set for <i>XPDMXML Project</i> for 3DEXPERIENCE data import from PDM Project.		
<b>Function:</b>	String <b>getProjectForV6</b> ( String          project, String          owner )		
<b>Class:</b>	CustomUserMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	String	project	the PDM Project
	String	owner	the PDM owner
	return		the XPDMXML Project to set
<b>Default:</b>	The default implementation returns the Project Name of the PDM project defined in TCI setting file <i>TCISettings.xml</i> . If no TCI setting is defined, an empty string is returned.  For further information about the TCI setting file, please refer to the chapter <i>TCI Configuration</i> in the TCI installation manual.		

Table 49: Retrieve value for Project for 3DEXPERIENCE import

To adapt the owning user for PDM import from 3DEXPERIENCE, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve the value for Owing User for PDM import from 3DEXPERIENCE.		
<b>Function:</b>	String <b>getOwningUserForV5</b> ( com.tsystems.tci.xpdmxml.XpdmBusinessObject                      busObject )		
<b>Class:</b>	CustomUserMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	XpdmBusinessObject	busObject	the XML object from 3DEXPERIENCE
	return		the owning user to set
<b>Default:</b>	The default implementation returns "null". (no user to set)		

Table 50: Retrieve the Owing User for PDM import

To adapt the owning group for PDM import from 3DEXPERIENCE, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve the value for Owing Group for PDM import from 3DEXPERIENCE.		
<b>Function:</b>	String <b>getOwningGroupForV5</b> ( com.tsystems.tci.xpdmxml.XpdmBusinessObject                      busObject )		
<b>Class:</b>	CustomUserMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	XpdmBusinessObject	busObject	the XML object from 3DEXPERIENCE
	return		the owning group to set
<b>Default:</b>	The default implementation returns "null". (no group to set)		

Table 51: Retrieve the Owing Group for PDM import



To adapt the list of additional objects which should be included in the change owner function for PDM import, implement the following extension point:

<b>Description:</b>	The interface function is called to retrieve a list of additional objects which should be included in the change owner function for PDM import.		
<b>Function:</b>	<code>IPdmModelObject[] <b>getAdditionalChangeOwnerObjects</b> (</code> <code>com.tsystems.tci.xpdmxml.XpdmBusinessObject</code> <code>busObject,</code> <code>com.tsystems.tci.pdm.interfaces.IPdmSession</code> <code>session,</code> <code>com.tsystems.tci.pdm.interfaces.IPdmModelObject</code> <code>pmo )</code>		
<b>Class:</b>	CustomUserMapping		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	<code>XpdmBusinessObject</code>	<code>busObject</code>	the XML object from 3DEXPERIENCE
	<code>IPdmSession</code>	<code>session</code>	the PDM Session
	<code>IPdmModelObject</code>	<code>pmo</code>	the PDM object
	<code>return</code>		the additional objects or null
<b>Default:</b>	The default implementation returns "null".		

Table 52: Retrieve list of additional objects for change owner function for PDM import

## 2.12 Customizing item reuse

To adapt the creation of a new reusable Item Revision in PDM, implement the following extension point:

<b>Description:</b>	The interface function is called for creation of a new reusable Item Revision in PDM. This Item will be created to reserve a Part Number.		
<b>Function:</b>	<pre>com.tsystems.tci.pdm.interfaces.IPdmModelObject <b>createReuseableItem</b> (     final com.tsystems.tci.customization.Customization customization,     final com.tsystems.tci.pdm.interfaces.IPdmSession session,     final String currentPartnumber )</pre>		
<b>Class:</b>	ItemReuse		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	final Customization	customization	the customization loader
	final IPdmSession	session	the Session to use
	final String	currentPartnumber	the proposed Part Number
	return		the created Item Revision
<b>Default:</b>	The default implementation returns a new Item Revision for reuse.		

Table 53: Creation of a new reusable Item Revision in PDM

To adapt the reading of a reusable Item Revision from PDM, implement the following extension point:

<b>Description:</b>	The interface function is called for reading of a reusable item revision from PDM.		
<b>Function:</b>	<pre>com.tsystems.tci.pdm.interfaces.IPdmModelObject <b>getReuseableItemRevision</b> (     final com.tsystems.tci.pdm.interfaces.IPdmSession session,     final com.tsystems.tci.pdm.interfaces.IPdmModelObject item )</pre>		
<b>Class:</b>	ItemReuse		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	final IPdmSession	session	the Session to use
	final IPdmModelObject	item	the Item to get usable Item Revision for
	return		the Item Revision to use
<b>Default:</b>	The default implementation returns "null".		

Table 54: Reading of a reusable Item Revision from PDM

## 2.13 Customizing PDM import

To adapt the preparation of newly created PDM Item for the import, implement the following extension point:

<b>Description:</b>	The interface function is called if a new Item (Product or Fastener) is prepared for PDM import.		
<b>Function:</b>	<pre>void prepareNewItem (     final com.tsystems.tci.pdm.interfaces.IPdmSession    session,     final com.tsystems.tci.pdm.interfaces.IPdmModelObject item )</pre>		
<b>Class:</b>	PDMImport		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	final IPdmSession	session	the PDM Session
	final IPdmModelObject	item	the Item to prepare
<b>Default:</b>	The default implementation performs no action.		

Table 55: Prepare newly created PDM Item for the import

To adapt the preparation of newly created PDM Item Revision for the import, implement the following extension point:

<b>Description:</b>	The interface function is called if a new Item Revision (Product or Fastener Instance) is prepared for PDM import.		
<b>Function:</b>	<pre>void prepareNewItemRevision (     final com.tsystems.tci.pdm.interfaces.IPdmSession    session,     final com.tsystems.tci.pdm.interfaces.IPdmModelObject item )</pre>		
<b>Class:</b>	PDMImport		
<b>Package:</b>	com.tsystems.tci.customization		
<b>Parameter:</b>	final IPdmSession	session	the PDM Session to use
	final IPdmModelObject	item	the Item to prepare
<b>Default:</b>	The default implementation performs no action.		

Table 56: Prepare newly created PDM Item Revision for the import



## 3 Customizing Example

The following example shows how to customize TCI for your specific company requirements. The content of this example is to adapt the default maturity mapping to specific needs.

Without any customizing TCI will read the maturity mapping for each project from the TCI settings and, in case of empty settings, using the default value (“SHARED”). If you want to adapt the 3DEXPERIENCE maturity to a specific value, a customizing is needed.

In this example, we like to adapt the maturity mapping to fulfill the following requirements:

- If a release state is already assigned to the object in PDM, set “FROZEN” for this object in 3DEXPERIENCE.
- If no release state is assigned to the object in PDM, set “IN\_WORK” for this object in 3DEXPERIENCE.

To customize the maturity mapping, the following steps have to be performed:

1. Create a new Java source file *MyMaturityMapping.java* inside a new java package with overriding method *getMaturity*. See chapter *Customizing lifecycle and maturity mapping* for detailed information about the method declaration of *getMaturity*.

Example code:

```
package com.my.tci.customization;
```

```
public class MyMaturityMapping extends CustomMaturityMapping
{
    @Override
    public String getMaturity(String project, String releaseStates)
    {
        if (releaseStates!= null && releaseStates.length(>0)
        {
            return "FROZEN";
        }
        else
        {
            return "IN_WORK";
        }
    }

    @Override
    public String getDescriptor()
    {
        return "My_" +getCustomizationType();
    }
}
```

2. Compile the Java source file *MyMaturityMapping.java*
3. Generate the jar file *MyCustomization.jar* with content

*MyMaturityMapping.class*

4. Copy the jar file *MyCustomization.jar* to your TCI lib directory

[TCI\_HOME][\apache-tomcat-dir]\webapps\XpdmAdapter\WEB-INF\lib

5. Register your customizing in the TCI setting:

```
<Customizations>
```

```
  <CustomizationPoint
```

```
    custpoint="com.tsystems.tci.customization.CustomMaturityMapping"  
    entrypoint="com.my.tci.customization.MyMaturityMapping"/>
```

```
</Customizations>
```

Further information about registration can be found in chapter *Customization registration* inside the TCI installation manual.

6. Test the customization example by transferring new data from PDM to 3DEXPERIENCE and checking the assigned states in 3DEXPERIENCE.