

Inhalt

GTL Data Importer	2
1. Konfiguration.....	2
2. Konfigurationsoptionen.....	2
3. Konfigurationsdatei	2
4. Alternative Konfigurationsdateien CFG.....	5
5. Überschreiben einzelner Konfigurationsoptionen	6
6. Notwendige Konfigurationsoptionen	6
7. Exkurs: Neue Bibliothek anlegen.....	7
8. Optional: WT Ordner als Kategorien importieren?	7
9. Allgemeine Voraussetzungen	10
9.1. Übertragung aus Windchill via SavedSearch und REST-API	11
9.1.1. Einrichten der Saved Search	12
9.1.2. Konfigurationsoptionen festlegen.....	14
9.1.3. Exkurs EncryptionTool	15
9.1.4. Rest-XSLT	18
9.1.5. Programmstart	20
9.2. Datenübertragung via RestApi (Alternative)	20
9.2.1. Einrichten der Konfigurationsoptionen.....	20
9.2.2. Swagger UI.....	22
9.2.3. Query Intervall und Zeitraum	26
9.2.4. XSLT-Datei.....	27
9.2.5. Programmstart	27
10. Übertragung aus Fremdsystemen	27
10.1. Übertragung via CSV.....	28
10.1.1. Regeldateien.....	28
10.1.2. Konfigurationsoptionen CSV	29
10.1.3. CSV mit Kopfzeile.....	29
10.1.4. CSV ohne Kopfzeile	30
2.2.1 Allgemein XML-Files aus Fremdsystemen	30
2.2.2 XML-Konfigurationsoptionen	30
2.2.3 XML-Files	31
2.2.4 XSLT-Files	31
11. Andere Dateiformate	32

GTL Data Importer

1. Konfiguration

GTL DataImporter kann über die Konfigurationsoptionen in der Datei „..\conf\main.cfg“ gesteuert.

Diese legen die Datenquelle fest, also welcher Dateityp ausgewertet werden soll bzw. ob es sich um den Import von Daten aus PTC Windchill handelt.

Sie bestimmen den Ablageort aller zur Verarbeitung relevanten Verzeichnisse, Regelfiles und der Log-Datei.

Weitere Konfigurationsoptionen bestimmen unterstützende Parameter zur Datenauswertung.

Sie findet sich im Installationsverzeichnis unter

<Laufwerkname:\Installationsordner>GT_Library_DataImporter\Software\conf\main.cfg
und lässt sich über einen beliebigen Texteditor verändern.

2. Konfigurationsoptionen

24 Einstellungen lassen sich über die Konfigurationsdatei vornehmen.

Sollten Datei oder Ordnerpfade modifiziert werden, ist darauf zu achten, dass ein entsprechender Ordner existiert, den Pfad anzugeben und dass GTL DI Schreib-und Leserechte im Ziel besitzt.

I.d.R. werden nicht alle Konfigurationsoptionen benötigt.

Welche Einstellungen vorgenommen werden müssen, ist abhängig von der Quelle der Daten, die ausgewertet werden sollen bzw. der Methode mit der Windchill-Objekte bezogen werden sollen.

Sollen in einem Durchlauf mehrere Quellen ausgewertet werden, müssen auch alle Konfigurationsoptionen dafür ausgefüllt sein.

Standardwerte sind hart vercodet und werden durch Werte aus einer CFG-Datei überschrieben, sie können auch durch Überschreibung in der task.cmd (..\GT_Library_dataImporter\task.cmd) geändert werden.

3. Konfigurationsdatei

Name	Verwendet von
gtl_dataimporter_db_path	Alle Varianten
GT Library Datenbankpfad inklusive Dateiname muss über CFG, task.cmd bzw. direkt aus Konsole übergeben werden Standardpfadmuster: Drive:\Folder_Structure\gt_resource_folder\library\DatabaseName.db	
gtl_dataimporter_log_dir	Alle Varianten
Aktuelle und letzte vorangegangene Logdatei mit Erfolgs- und Fehlermeldungen werden hier gespeichert; es wird empfohlen, den Standardpfad beizubehalten, um eventuelle Konflikte mit Schreibrechten zu vermeiden. Name der Datei: log.txt	

<p>Vorgänger wird in oldLog.txt umbenannt. Der Name ist nicht bei der Pfaddefinition anzugeben. Der Pfad ist als absoluter Pfad oder relativ zu DataImporter.exe anzugeben. Wird er nicht gesondert definiert, ist er relativ zu DataImporter.exe unter ../../data/log zu finden. Standardpfad und verwendeter Pfad bis zur endgültigen Konfiguration durch Startargumente bzw. CFG-Dateien: Drive:\Folder_Structure\GT_Library_DataImporter\data\log Der Ordnerpfad darf geändert werden, der Originalordner sollte unabhängig davon nicht gelöscht werden</p>	
gtl_dataimporter_polling_dir	Alle Varianten
<p>Sämtliche Dateien mit zur Übertragung vorgesehenen Objekten werden hier zur Prüfung abgelegt. Der Pfad ist als absoluter Pfad oder relativ zu DataImporter.exe anzugeben. Wird er nicht gesondert definiert, ist er relativ zu DataImporter.exe unter ../../data/polling zu finden. Standard: Drive:\Folder_Structure\GT_Library_DataImporter\data\polling Der Ordnerpfad darf geändert werden, der Originalordner sollte unabhängig davon nicht gelöscht werden</p>	
gtl_dataimporter_success_dir	Alle Varianten
<p>Alle erfolgreich verarbeiteten Dateien werden hierhin verschoben von GTLDI. Der Pfad ist als absoluter Pfad oder relativ zu DataImporter.exe anzugeben. Wird er nicht gesondert definiert, ist er relativ zu DataImporter.exe unter ../../data/success zu finden. Standard: Drive:\Folder_Structure\GT_Library_DataImporter\data\succes Der Ordnerpfad darf geändert werden, der Originalordner sollte unabhängig davon nicht gelöscht werden</p>	
gtl_dataimporter_error_dir	Alle Varianten
<p>Alle Dateien mit nicht übertragbaren Objekten werden hierhin verschoben von GTLDI. Der Pfad ist als absoluter Pfad oder relativ zu DataImporter.exe anzugeben. Wird er nicht gesondert definiert, ist er relativ zu DataImporter.exe unter ../../data/error zu finden. Standard: Drive:\Folder_Structure\GT_Library_DataImporter\data\error Der Ordnerpfad darf geändert werden, der Originalordner sollte unabhängig davon nicht gelöscht werden</p>	
gtl_dataimporter_sleep_time_int	Alle Varianten
<p>Wartezeit in Sekunden zwischen Speicherung der zu prüfenden Dateien in polling und deren tatsächlicher Verarbeitung, um Zeitverzögerung bei automatisierter Speicherung der Prüfdaten auszugleichen. Sollte nichts gesondert übergeben werden, oder eine Einstellung mit nicht definiertem Wert, wird von DataImporter automatisch 2000 gesetzt</p>	
gtl_dataimporter_change_parameter	Alle Varianten*
<p>Name des Parameters, anhand dessen geprüft werden soll, ob ein Objekt verändert wurde. Ein korrespondierender Parameter muss in übertragenen Objekten vorhanden sein, um die Überschreibung unveränderter oder neuerer Datenbankwerte zu verhindern. In der Regel eine Datumsangabe. Unterschiedliche Datumsformate können zu Überschreibung neuerer Werte führen. Es wird ein eindeutiger betriebsinterner Standard empfohlen. Standard in main.cfg P:CHANGEDATE</p>	
gtl_dataimporter_db_thumbnail_path	Alle Varianten*
<p>Absoluter oder relativer Pfad zum Bilddateiordner der Datenbank. Standardpfadmuster: Drive:\Folder_Structure\gt_resource_folder\library\DatabaseNameFolder\ Darin enthalten img, img_w40 und img_w100; werden, wenn nicht vorhanden automatisch angelegt. Notwendig sollen Vorschaubilder eines Objektes an GT Library übergeben werden</p>	
gtl_dataimporter_wc_server	Alle WC-Lösungen
<p>Name des Windchill-Servers eingebettet in Teil URL mit /Windchill am Ende z.B. http://ServerAddress:Port/Windchill</p>	
gtl_dataimporter_wc_login_name	Alle WC-Lösungen
<p>Windchill Login-Name; muss für Windchilllösungen unverschlüsselt oder verschlüsselt(s.u.) vorliegen; im Zweifel bevorzugt DataImporter die verschlüsselte Variante</p>	

gtl_dataimporter_wc_login_password	Alle WC-Lösungen
Windchill Login_Passwort muss unverschlüsselt oder verschlüsselt (s.u.) vorliegen	
gtl_dataimporter_wc_login_name_encrypted	Alle WC-Lösungen
Windchill Login-Name mit GTL_DataImporter Encryption Tool verschlüsselt	
gtl_dataimporter_wc_login_password_encrypted	Alle WC-Lösungen
Windchill Login_Passwort mit GTL_DataImporter Encryption Tool verschlüsselt	
gtl_dataimporter_wc_rest_xslt	Alle WC-Rest-API Lösungen
Absoluter oder relativer Pfad zur XSLT-Datei zur Verarbeitung von Daten aus Windchill, die via REST-API bezogen wurden, ein kommentiertes Beispiel findet sich unter Drive:\Folder_Structure\GT_Library_DataImporter\software\conf\rest.xslt	
gtl_dataimporter_rest_select	Rest mit WC-Saved Search
Angabe aller Attribute die bei den übergebenen Objekten ausgewertet werden sollen. Die Werte sind mit Komma getrennt ohne Leerzeichen einzutragen. Das empfohlene Beispiel welches mit der Beispiel rest.xslt korrespondiert ist: CADName,container,folder,state,version,MATERIAL,DESCRIPTION_1_DE,DESCRIPTION_2_DE,DESCRIPTION_1_EN,DESCRIPTION_2_EN,thePersistInfo.modifyStamp	
gtl_dataimporter_wcSearch_name	Rest mit WC-Saved Search
Name der Saved Search aus Windchill, die verwendet werden soll	
gtl_dataimporter_wc_rest_query	WC Rest-Abfrage ohne Saved Search
Rest URL für Abfragen, die nicht auf einer Saved Search basieren. Inhalt entspricht den Elementen dem SELECT der oben aufgeführten Abfrage Bestandteile: <ul style="list-style-type: none"> - http:// - Address:Port - /Windchill/servlet/rest/structure/objects - ?%24filter= - (endswith(name%2C%20'prt')%20or%20endswith(name%2C%20'asm')%20%20or%20endswith(name%2C%20'drw')%20or%20endswith(name%2C%20'gph')%20or%20endswith(name%2C%20'mfg')%20or%20endswith(name%2C%20'lay')%20or%20endswith(name%2C%20'frm')%20or%20endswith(name%2C%20'tbl')%20or%20endswith(name%2C%20'sym')%20or%20endswith(name%2C%20'txt')%20or%20endswith(name%2C%20'sec'))%20and%20state%20eq%20'RELEASED' - &%24select=CADName%2Ccontainer%2Cfolder%2Cversion%2CMATERIAL%2CDESCRIPTION_1_DE%2CDESCRIPTION_2_DE%2CDESCRIPTION_1_EN%2CDESCRIPTION_2_EN%2CthePersistInfo.modifyStamp - &typeld=wt.epm.EPMDocument - &queryLimit= 	
gtl_dataimporter_rest_query_int	WC Rest-Abfrage ohne Saved Search
Int-Wert; Bestimmt die Menge der Objekte, die pro Restaufruf geholt werden. Bei wenig zu erwartenden Objekten eher klein zu wählen, bei vielen erwarteten Objekten besser größer zu wählen. Wert 0 wählt das größtmögliche Intervall, also alle Objekte aus dem in Windchill ausgewählten Bereich.	
gtl_dataimporter_rest_query_days	WC Rest-Abfrage ohne Saved Search
Int-Wert; Anzahl der zurückliegenden Tage, aus denen Objekte abgerufen werden sollen. Wert 0 wählt größtmöglichen Zeitraum	
gtl_dataimporter_wc_webapi_xslt	WC Abfrage mit webAPI
XSLT-Datei zum Auswerten von Objekten, die via Web-API/JSP-Action-Call aufgerufen wurden; Verwendung nicht empfohlen	
gtl_dataimporter_xml_rule_file	Xml (Fremdsysteme)
XSLT-Datei für XML aus anderen Systemen als Windchill, z.B. Drive:\Folder_Structure\GT_Library_DataImporter\software\conf\xml_rule.xslt	
gtl_dataimporter_csv_rule_file	csv(Fremdsysteme)

Anweisungsdatei, txt-Format, bestimmt die Attribute und deren Position in den auszuwertenden CSV-Dateien. Abhängig vom Aufbau des CSV werden die Begriffe der Kopfzeile oder die ,Positionen der Attribute zugeordnet Standards in Drive:\Folder_Structure\GT_Library_DataImporter\software\conf\csv_rule_header.txt Bzw. Drive:\Folder_Structure\GT_Library_DataImporter\software\conf\csv_rule_headerless.txt	
gtl_dataimporter_csv_header_bool	csv(Fremdsysteme)
Bool-Wert (0/1) oder false/true); Gibt an, ob die erwarteten csv-Dateien eine Kopfzeile haben (1/true) oder nicht (0/false)	
gtl_dataimporter_csv_separator_char	csv(Fremdsysteme)
Char-Wert; gibt das verwendete Trennzeichen der erwarteten csv-Datei(en) an. I.d.R Komma(,) oder Semikolon (;)	

*optional; empfohlen aber nicht zwingend erforderlich

Siehe auch Standard-CFG-Datei main.cfg unter

Drive:\Folder_Structure\GT_Library_DataImporter\software\conf

Eine Datei main.cfg unter „..\software\conf“ muss zwingend existieren.

Sie kann direkt mit den nötigen Konfigurationen befüllt werden.

Sie dient gleichzeitig als Vorlage für alternative CFG-Dateien.

Sie enthält alle möglichen Konfigurationsoptionen mit Kurzerklärung.

Eine Konfigurationsdatei muss die Dateiendung .cfg aufweisen.

Jede Zeile in einer CFG-Datei folgt dem Muster:

gtl_dataimporter_schluesel = Wert

Zwischen Schlüssel und Wert darf kein Zeilenumbruch erfolgen.

Anmerkungen werden am Zeilenanfang mit einem Semikolon gekennzeichnet:

;Anmerkungstext

Leerzeilen werden von GTL_DataImporter ignoriert

Nicht benötigte Zeilen können mit einem Semikolon auskommentiert werden.

4. Alternative Konfigurationsdateien CFG

Sollten verschiedene Konfigurationen von GTL DI häufiger benötigt werden, bietet sich an, mehrere CFG-Dateien vorrätig zu halten.

Um diese in das Programm einzubinden, werden sie entweder via Konsolenbefehl oder in der task.cmd-Datei übergeben.

Ein Pfad zu einer Konfigurationsdatei wird **direkt hinter** dem Programmaufruf als absoluter Pfad oder relativ zu DataImporter.exe übergeben, dem „-p“ vorangestellt ist:

„-p Drive:\path\CFGFile.cfg>“

Sämtliche in der CFG-Datei enthaltenen Werte werden modifiziert. Nicht verwendete Werte verbleiben auf Standard oder wie zuvor von main.cfg übernommen.

Beispiel der task.cmd:

@echo off

cd software/bin/

DataImporter.exe -p C:\daten\svn\gtl-data-importer\trunk\GT_Library_DataImporter\software\conf\alternative.cfg

5. Überschreiben einzelner Konfigurationsoptionen

Einzelne Werte können direkt verändert werden.

Dies erfolgt über Konsole oder CMD-Datei. Konfigurationsoptionen werden mit

„-c gtl_dataimporter_config_name Wert“ hinter dem Programmaufruf oder dem Aufruf einer alternativen CFG_Datei übergeben.

Bsp.:

@echo off

cd software/bin/

DataImporter.exe -p ..\GT_Library_DataImporter\software\conf\alternative.cfg -c gtl_dataimporter_config_name Wert

Werden mehrere konkurrierende Modifikation an den Konfigurationsoptionen vorgenommen, gilt die zuletzt übergebene Modifikation.

Unter -p wird in einer CFG-Datei z.B. unter gtl_dataimporter_db_path ein Datenbankpfad festgelegt. Mit der Anweisung unter -c wird dieser Pfad mit einem anderen überschrieben.

Alle nicht modifizierten Werte, bleiben automatisch auf Standard bzw. wie von main.cfg übernommen.

6. Notwendige Konfigurationsoptionen

- gtl_dataimporter_log_dir
- gtl_dataimporter_polling_dir
- gtl_dataimporter_success_dir
- gtl_dataimporter_error_dir
- gtl_sleeptime_int

haben einen hart vercodeten Wert,

können geändert werden (direkt in main.cfg, via -c oder -p; siehe oben)

bei log nicht empfohlen.

Auf Schreib-/Leserechte für GTL DI achten bei Verlegung!

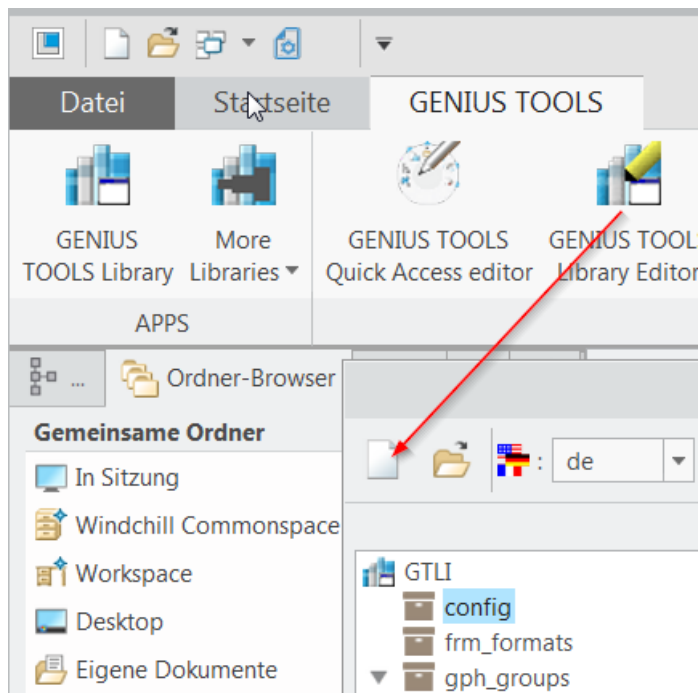
- gtl_dataimporter_db_path

Legt den Pfad zur GT Library Datenbank fest.

Ist nicht hart vercodet und muss zu Programmstart übergeben werden.

Der entsprechende Pfad wird beim Einrichten der GT Library Bibliothek automatisch angelegt und kann von dort kopiert und in die CFG eingefügt werden.

7. Exkurs: Neue Bibliothek anlegen



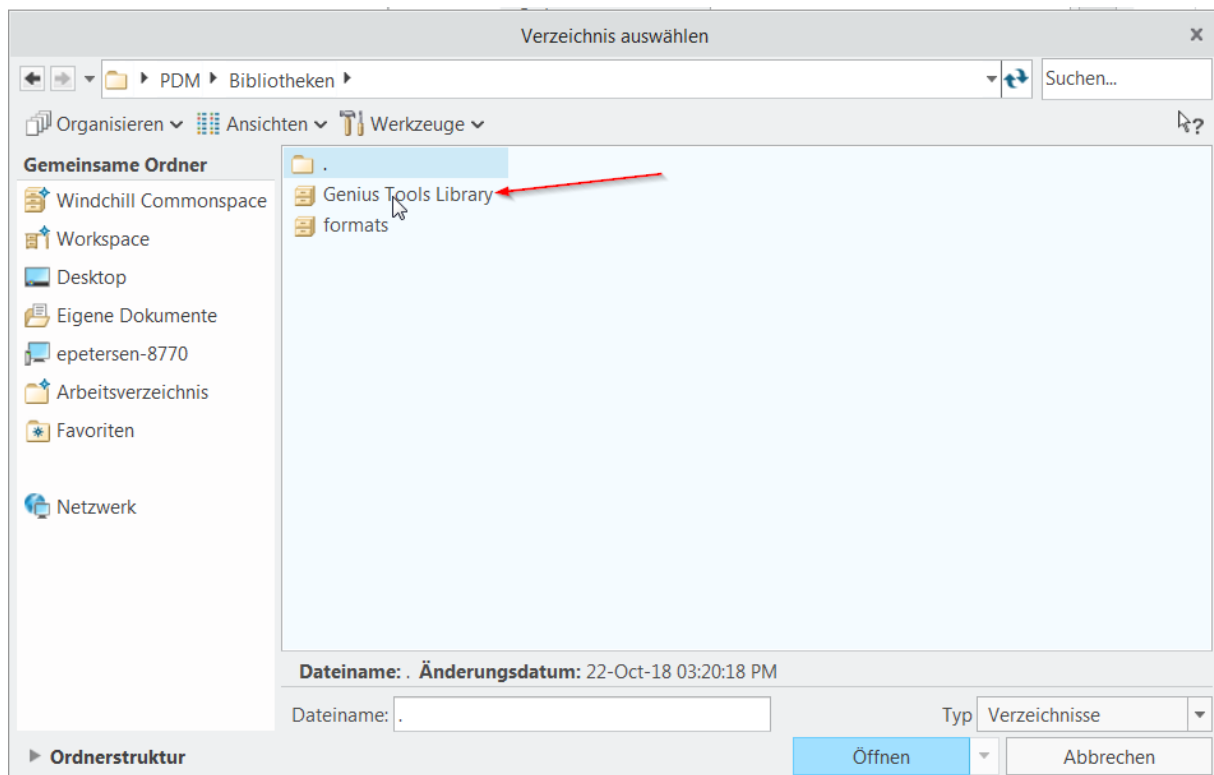
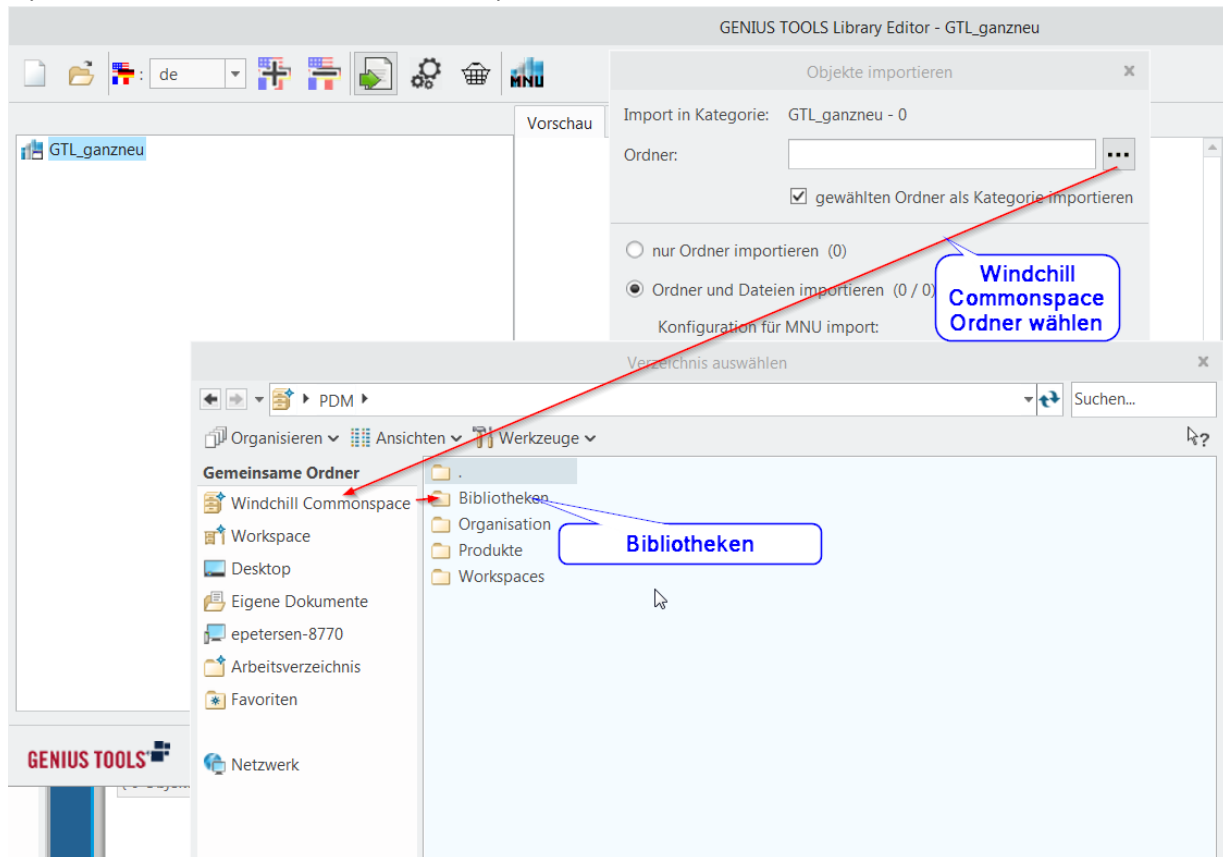
Namen eingeben

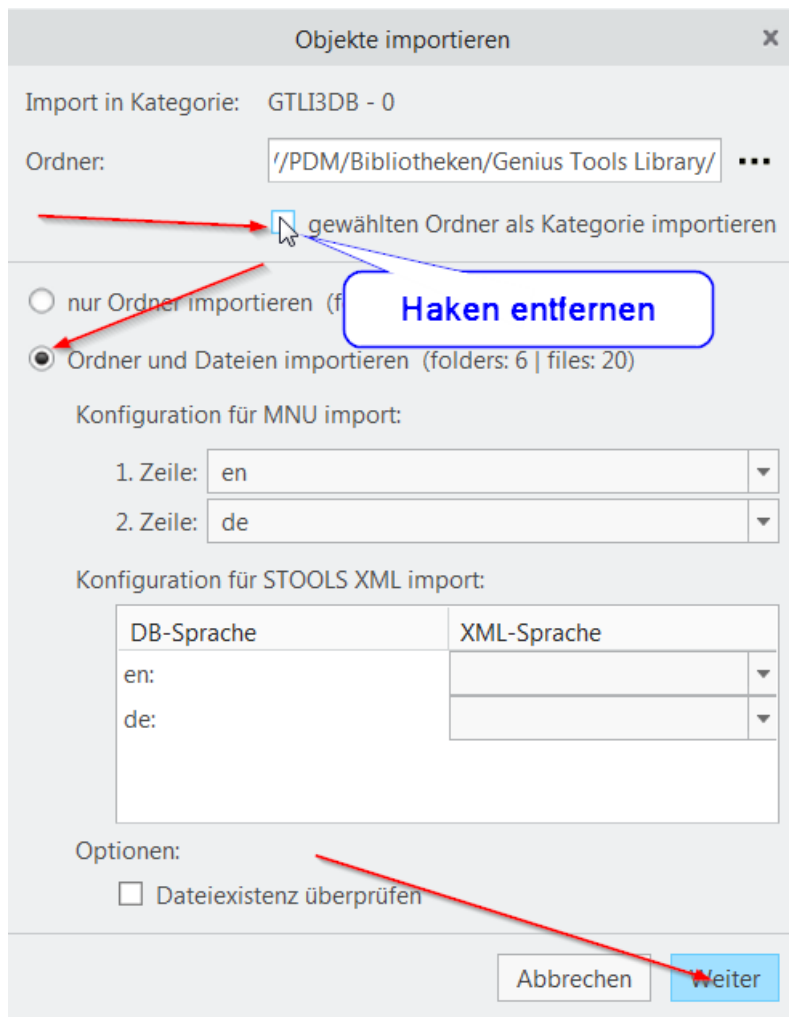
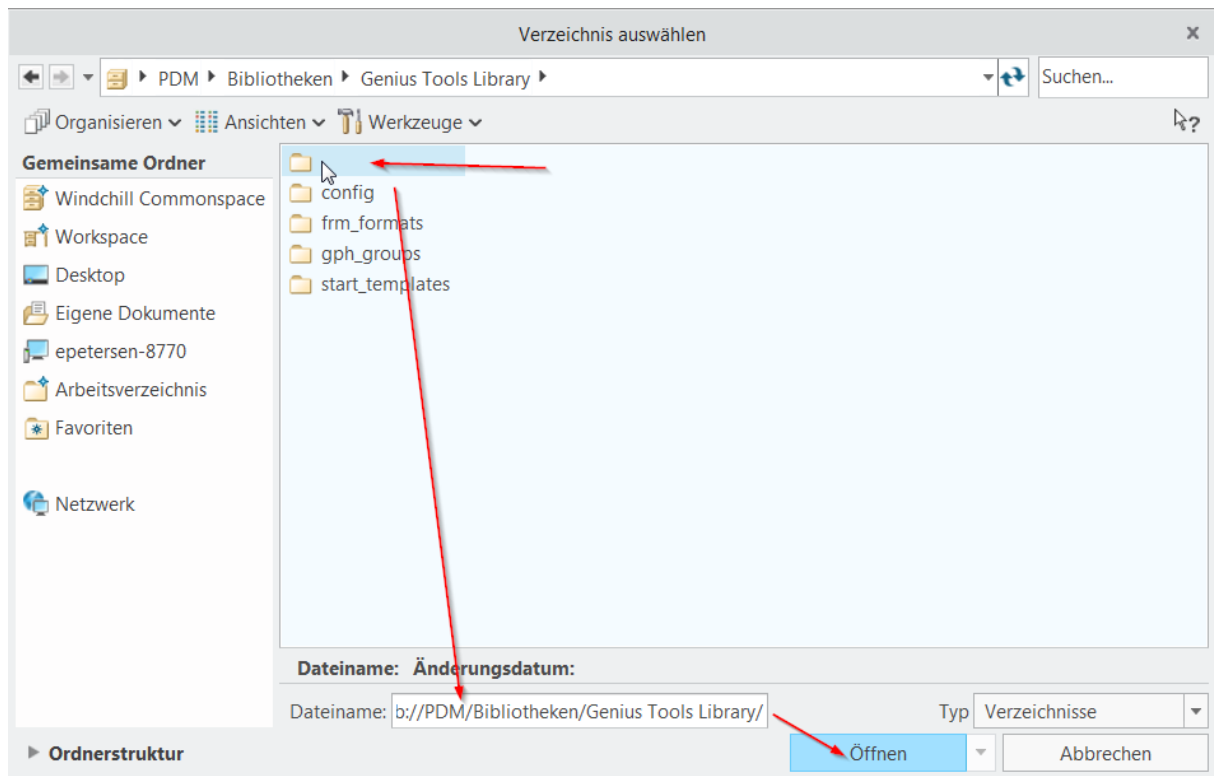
OK drücken

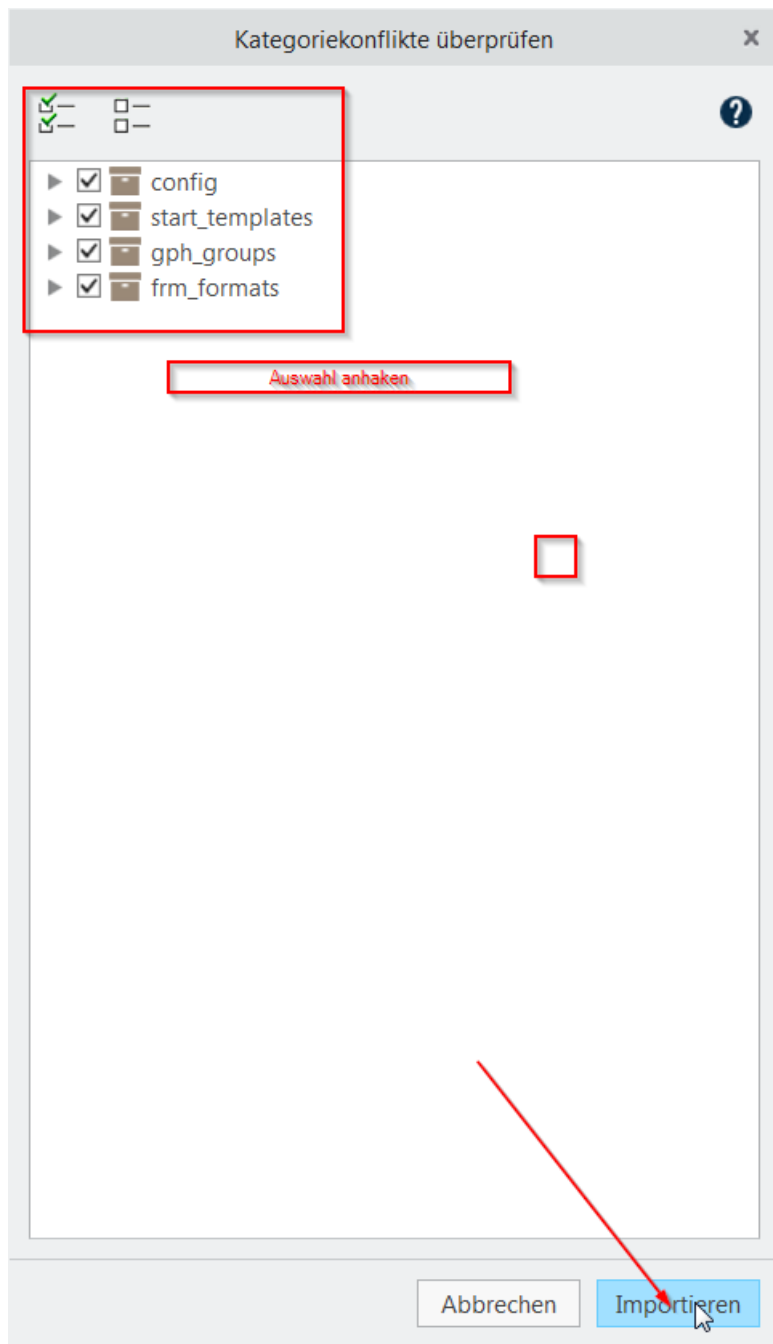
➔ Neue Bibliothek wird angelegt und geöffnet

8. Optional: WT Ordner als Kategorien importieren?

Optional: Ordner UND DATEIEN initial importieren







9. Allgemeine Voraussetzungen

Für den Betrieb von GTL_DI sind verschiedene Hilfsdateien nötig. Neben der CFG-Datei sind diese abhängig vom gewählten Übertragungsformat.

Sämtliche Regeldateien befinden sich als Beispiele/Vorlagen unter

Drive:\Folder_Structure\GT_Library_DataImporter\software\conf

Grundsätzlich unterscheiden sich zwei Arten:

- Übertragungsformate aus Fremdsystemen
- Übertragung aus Windchill

Die einzelnen Konfigurationen werden in den jeweiligen Unterpunkten beschrieben.

Mögliche zu verarbeitende Dateien finden sich unter

Drive:\Folder_Structure\GT_Library_DataImporter\software\PollingExamples

Das Attribut „name“ muss immer übergeben werden.

Direkt übergeben werden können auch:

- path
- objType
- info
- status

Bilder/Thumbnails können im base64-Format unter „thumbnail“ übergeben werden.

Bei Daten aus PTC Windchill wird hingegen die entsprechende ID des Objektes hinterlegt z.B:

OR:wt.epm.EPMDocument:29758012

Zusätzliche Parameter werden übergeben als

P:PARAMETERNAME

Dimensionen:

D:DIMENSIONSNAME

Objektbezeichnungen in verschiedenen Sprachen z.B.

title_de oder title_en

Kommen mehrere Objekte aus derselben Quelle (z.B. aus Windchill oder einer CSV-Datei mit mehreren Zeilen) und es tritt bei der Vorbereitung zur Datenbankübertragung ein Fehler auf, wird die komplette Datei in den Error-Ordner verschoben und es findet keine Übertragung in die Datenbank statt.

Sind mehrere Objekte mit identischem Namen in der zu übertragenden Datei, wird ebenfalls ein Fehler ausgegeben.

Sämtliche Objekte, außer die mit der Namensdopplung, werden allerdings in die Datenbank übertragen. Die entsprechende Datei, wird dennoch in den Error-Ordner verschoben. In allen Fällen erfolgt ein Hinweis im Logfile. Auf eine Namensdopplung wird noch einmal konkret am Ende des Logfiles hingewiesen.

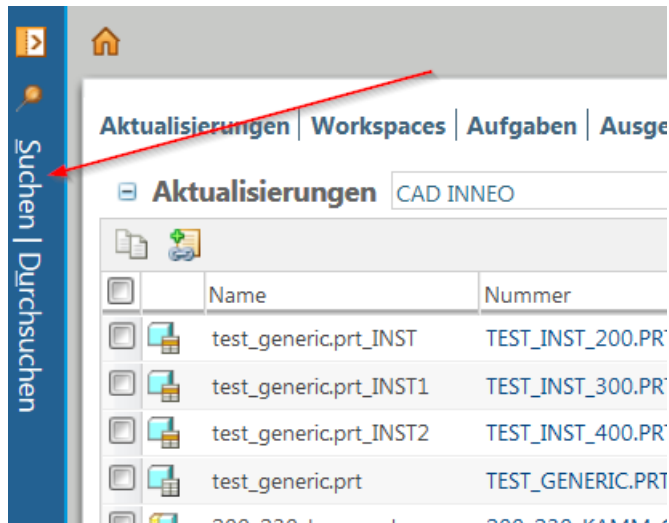
Dateien, die sich am Ende des Importprozesses noch im Polling-Ordner befinden, werden in den Error-Ordner verschoben.

Alle Dateien, deren Datensätze erfolgreich übertragen wurden, werden in den Success-Ordner verschoben.

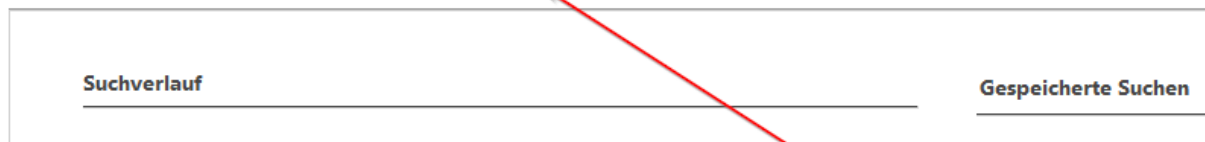
9.1. Übertragung aus Windchill via SavedSearch und REST-API

Empfohlene Methode für alle Windchillnutzer.

9.1.1. Einrichten der Saved Search



Suchverlauf und gespeicherte Suchen | Erweiterte Suche



Suchverlauf und gespeicherte Suchen | Erweiterte Suche

Suchbegriff:

Typ

☒ Alle Typen

☐ Meine Favoritentypen [Hinzufügen](#)

☐ CAD-Dokument ☐ Dokument

Kontext

☐ Alle Kontexte [Weitere Optionen](#)

☐ Meine Favoritenkontexte [Hinzufügen](#)

☐ Genius Tools Li... ☒ Maschinenbau

Kriterien

()

UND

Name =

Nummer =

Favoriten auswählen erst Typen (oben, eventuell Dokument „enthaken“), dann Kontexte (unten)

Suchbegriff:

Typ

☐ Alle Typen

☐ Meine Favoritentypen [Hinzufügen](#)

☒ CAD-Dokument ☐ Dokument

Kontext

☐ Alle Kontexte [Weitere Optionen](#)

☒ Meine Favoritenkontexte [Hinzufügen](#)

☒ Genius Tools Li... ☒ Maschinenbau

Kriterien

()

UND

Name =

Nummer =

Als nächstes Kriterien festlegen

Kriterien

((Lebenszyklusstatus = 'Freigegeben') UND (Letzte Änderung = 'Von:5 Bis:0') UND (Revision = 'Neueste'))

UND

Lebenszyklusstatus Freigegeben

Letzte Änderung Tagesbereich auswählen Von: 5 vor x Tagen--> Bis: 0 Tage ab dem heutigem Datum

Revision = ☒ Auswählen ☐ Bestimmen:

[Such](#)

In die ersten beiden Dropdown-Menüs:

Lebenszyklusstatus (falls z.B. nur freigegebene Gewünscht)

Letzte Änderung (für den gewünschten Zeitraum; bei regelmäßigen automatischen Importen nach Library den Zeitraum so wählen, dass es beim automatisierten Prozess nicht zu Überlappungen kommt)

Für weitere Auswahlkriterien neue Dropdown-Menüs erstellen:

Über Attribut zufügen neuen Reiter erstellen (z.B. Revision und dort bei Auswählen neuste)

Anschließend Gruppe hinzufügen und darin Attribute mit Oder verknüpfen, um die gewünschten Dokumententypen auszuwählen.

UND

Lebenszyklusstatus = Freigegeben

Letzte Änderung = Tagesbereich auswählen Von: 5 vor x Tage

Revision = Auswählen: Neueste Bestimmen:

ODER

Dokumentkategorie = CAD-Teil

Dokumentkategorie = Baugruppe

Dokumentkategorie = Zeichnung

Dokumentkategorie = Fertigung

Dokumentkategorie = Layout

Dokumentkategorie = Skizze

Dokumentkategorie = Notiz

Dokumentkategorie = Zeichnungsvorlage

Dokumentkategorie = Diagramm

Dokumentkategorie = Benutzerdefinierte Funktion

Dokumentkategorie = Format

Schließlich auf „Diese Suche speichern“ gehen und Namen für die Suche vergeben

9.1.2. Konfigurationsoptionen festlegen

Zu übergebende Konfigurationsoptionen, siehe Tabelle

Die Beispieldaten ergeben sich teilweise aus den Standards von GT_Library und Windchill.

Notwendige, gleichbleibende Strukturen in den Werten sind farbig markiert, der Rest sind Beispiele, die vom Kundensystem abhängen.

configOption	Beispielwert
gtl_dataimporter_db_path =	Drive:\path\gt_resource_folder\library\GTL_db.db
gtl_dataimporter_db_thumbnail_path =	Drive:\path\gt_resource_folder\library\GTL_db\
gtl_dataimporter_wc_server =	http://WT-Server:PortNr/Windchill
gtl_dataimporter_wc_login_name =	wcadmin
gtl_dataimporter_wc_login_password =	wcadmin
gtl_dataimporter_wc_login_name_encrypted =	lq3V,2T@E9?{[mHtBf9]24.]<z2cN)}Tp&?U9zq/H{@5O(#0yRE&+lQr"3G?nU<Tud9"8TiIOmti?0c6PXT%TQo7}Nu(jJ.4r?- ,W(t0}NOQ951\L)mIE WnIQ>UTZ\xy*Q(\$P&WS&Cdfy_cB7,:wc)(@X%t_Wap&>x\s;JE@E2Viajs3">dP }v)e>e\>r"a\8fm,Fd%:<u]m<D:L5:H0#r2%>'*N[RGZ2mAD\Z#X[i<oW{_:]Lm;m<]?IX
gtl_dataimporter_wc_login_password_encrypted =	+4Tv[QOZK_S9qY/VY5NW;M8V4yft9ps[{A3qFrS>[crmfyiO07Eq3wIA2J8FotM?t8J0lrM/*mR?HqIXS3A2R9<GW9huMGi&(oH?9AUt[UjRo

	%ALQRIZOW3iv5mUJ/7IV9pN&M4FnO8}Zod U6vZuF7HV/l}yi/GPtXO\rXheNXMNfQ&Bb&u otNgXzj[9qoY{syJ7 oL[Xf*Xu4gU@qk20"w{C KdC*Psq/8uy86R}1r20q"1;a*x/R!g@x&Cs4\$/ y
gtl_dataimporter_wc_rest_xslt =	..\conf\rest.xslt
gtl_dataimporter_change_parameter =	CHANGEDATE
gtl_dataimporter_rest_select =	CADName,container,folder,version,MATERIA L,DESCRIPTION_1_DE,DESCRIPTION_2_DE,D ESCRIPTION_1_EN,DESCRIPTION_2_EN,theP ersistInfo.modifyStamp
gtl_dataimporter_wcSearch_name =	GTLI2

Die Logindaten "gtl_dataimporter_wc_login_name =" und "gtl_dataimporter_wc_login_password =" sollten einem Account entsprechen, der Zugriff auf die abzurufenden Daten hat.

Die Logindaten "gtl_dataimporter_wc_login_name_encrypted =" und "gtl_dataimporter_wc_login_password_encrypted =" erfüllen dieselbe Funktion, werden allerdings von DataImporter bevorzugt zu ihren unverschlüsselten Pendanten ausgewertet. Sind sowohl verschlüsselter, als auch unverschlüsselter Wert ausgefüllt, überschreibt die entschlüsselte Variante, die unverschlüsselte im Programmverlauf.

Lässt sich ein Wert aus Gründen nicht entschlüsseln, versucht DataImporter die unverschlüsselte Variante zu verwenden.

Nur wenn beide ungültig oder nicht vorhanden sind, bricht DataImporter den Vorgang ab. Ebenso wenn Passwort und Nutzer zwar beide gültig aber nicht zusammengehörig sind. Darum sollte unbedingt darauf geachtet werden, dass entweder nur je eine Variante von Username oder Passwort verwendet wird, bzw. die verschlüsselte der unverschlüsselten entspricht. Es ist aber möglich das Passwort verschlüsselt, den Usernamen aber unverschlüsselt zu übergeben (oder umgekehrt).

Unter „gtl_dataimporter_wcSearch_name =" wird der Name der Suche hinterlegt.

Die Attribute zum Abruf eintragen unter „gtl_dataimporter_rest_select ="

Achtung, keine Leerzeichen zwischen die Attribute und jedes mit einem Komma trennen.

Um mögliche Attribute zu ermitteln bieten sich Listen an(z.B. in Creo unter Werkzeuge->Parameter, wenn ein Objekt geladen ist); weitere Möglichkeiten sind die Swagger UI in Windchill und entsprechende Abfragen oder die Anwendung Postman, mit denen sich REST-Abfragen testen lassen)

Als Wert für die letzte Änderung eines Objektes bietet sich „thePersistInfo.modifyStamp“ an. Dieses Attribut wird in der XSLT (siehe unten) dem entsprechenden Parameter (im Beispiel: P:

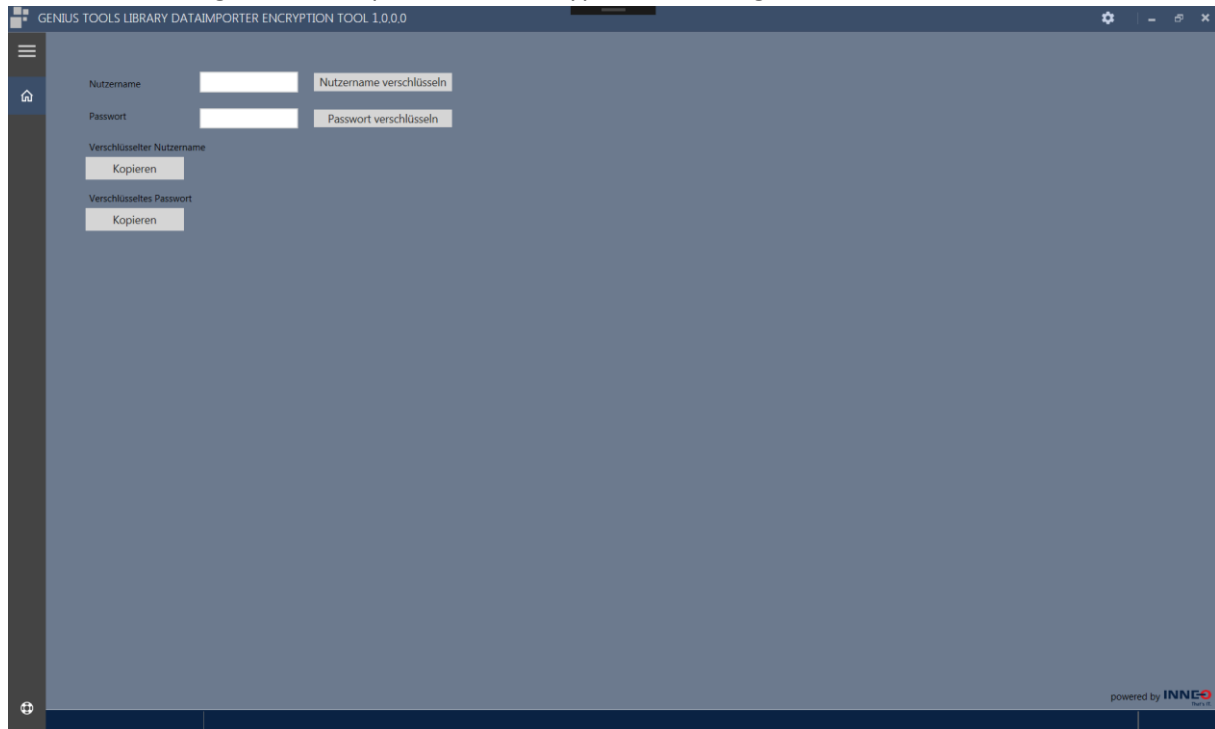
CHANGEDATE) zugeordnet.

9.1.3. Exkurs EncryptionTool

Für die Übergabe von Windchillpasswort und Username bietet sich an, diese zu verschlüsseln, damit sie nicht von unbefugten Dritten gelesen werden können.

Dafür wird das EncryptionTool bereitgestellt, welches Passwort und/oder Username so verschlüsselt, dass DataImporter beide auswerten kann.

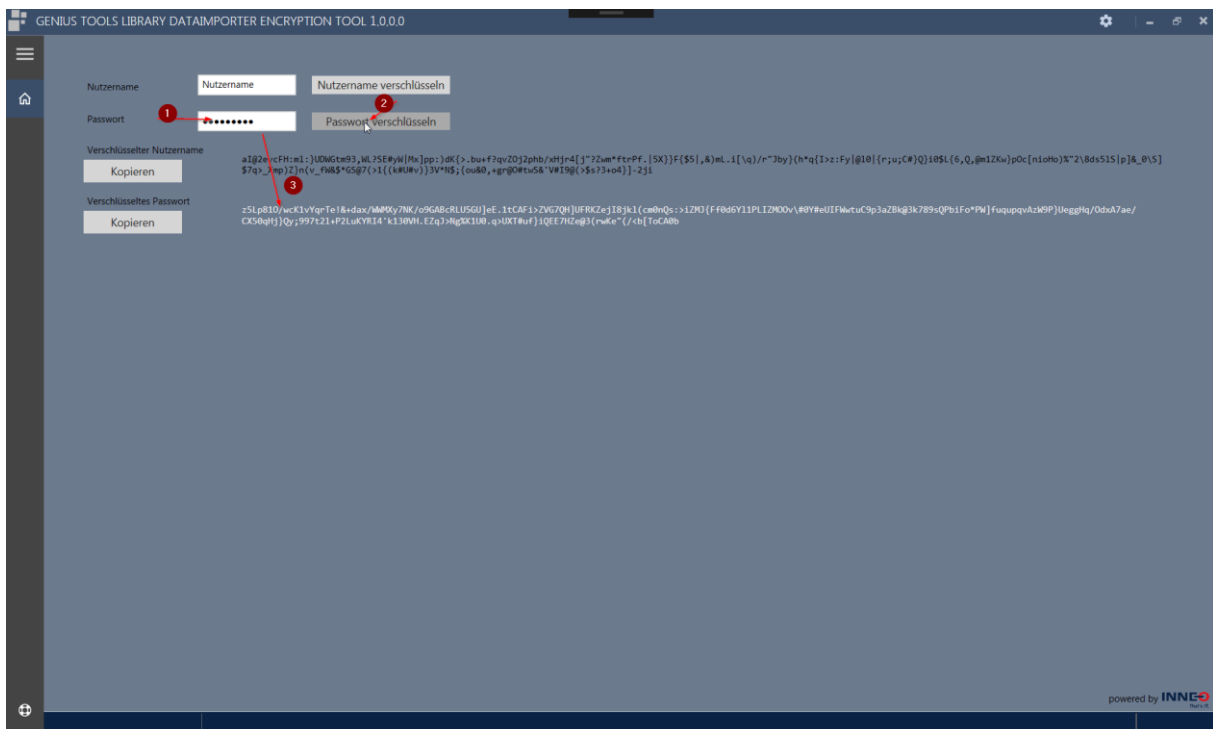
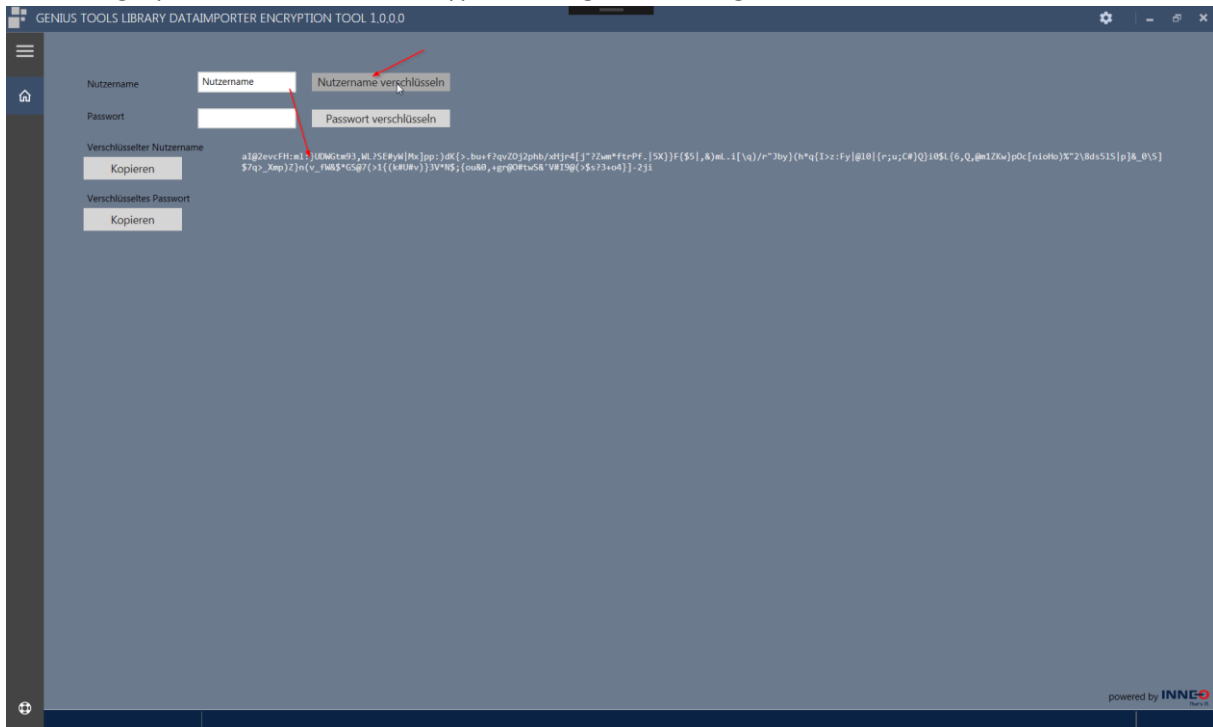
Es wird unabhängig von DataImporter über EncryptionTool.exe gestartet.



In Nutzername und Passwort können die entsprechenden Daten unabhängig voneinander eingegeben werden. Das Passwort wird generell nicht im Klartext auf dem Bildschirm angezeigt. Jedes hat einen eigenen Verschlüsselungbutton, der gedrückt werden kann, sobald das entsprechende Feld den vollständigen gewünschten Text enthält.

Es erscheint im Anschluss der verschlüsselte Wert und wird unmittelbar in den Zwischenspeicher gelegt. So kann er umgehend in die CFG_Datei an entsprechender Stelle eingefügt werden. Sollten beide Werte verschlüsselt werden, ist darauf zu achten, dass immer der Wert zum letzten gedrückten Button in den Zwischenspeicher genommen wird. Zu diesem Zweck wird der aktuell

zwischenengespeicherte Wert von EncryptionTool gelb hinterlegt.



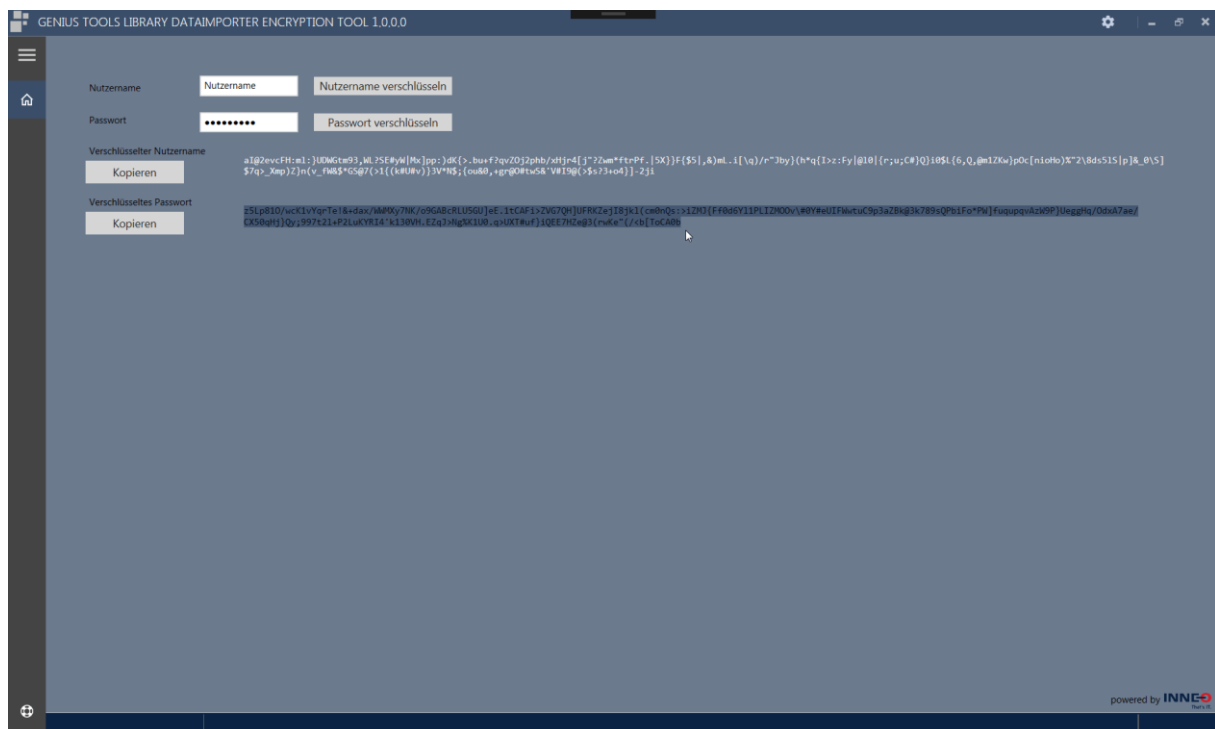
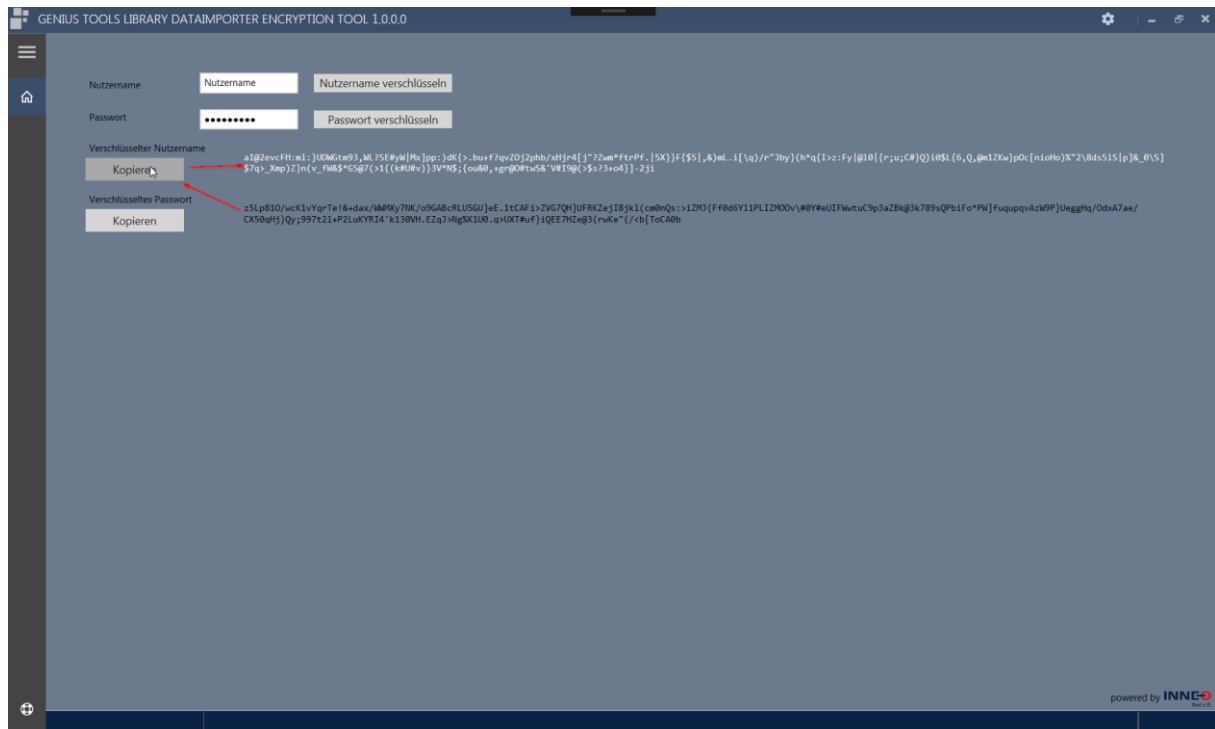
Nochmaliges Drücken eines Knopfes erzeugt eine neue Verschlüsselung, selbst wenn der Wert nicht geändert wurde (in diesem Fall, muss der neue Schlüssel aber NICHT noch einmal in die CFG-Datei übertragen werden.)

Sollte der Text in einem Feld verändert werden, wird auch das zugehörige Schlüsselfeld gelöscht und der Zwischenspeicher geleert. Mit erneutem Drücken des Encryptbuttons wird ein neuer Schlüssel generiert.

Soll der Wert im Zwischenspeicher nochmal gewechselt werden (z.B. der Username nach Erzeugung eines generierten Passwords noch einmal kopiert werden, lässt sich der nebenstehende Copy-Button

verwenden.

Alternativ lassen sich die Schlüssel auch händisch markieren und via CTRL-C in den Zwischenspeicher kopieren.



9.1.4. Rest-XSLT

Eine kommentierte Beispieldatei findet sich unter
Drive:\Folder_Structure\GT_Library_DataImporter\software\conf\rest.xslt

Generelle Anweisungen für alle XSLT-Dateien:

Alle Knoten müssen in der XSLT eine Zuweisung nach dem Muster `Attributsname=` erfahren, dem der Wert eines oder mehrerer adäquate XML-Knoten zugewiesen wird.

Alle Attribute müssen am Ende die Anweisung

```
<xsl:text>&#xa;</xsl:text>
```

erhalten.

Werden mehrere Objekte innerhalb einer Datei erwartet, sollte die XSLT-Datei eine `for-each`-Anweisung um die entsprechenden Knoten herum enthalten. Z.B.:

```
<xsl:for-each select="root/element/items"> </xsl:for-each>
```

Die letzte Anweisung für ein Objekt muss immer

```
<xsl:text>-&#xa;</xsl:text>
```

sein, damit DataImporter bei der späteren Auswertung das Ende eines Objektes erkennt.

Die zugehörige XML-Datei wird automatisch aus dem über Rest bezogenen JSON-File erzeugt und verarbeitet, muss also nicht generiert oder bearbeitet werden. Beide lassen sich nach Bearbeitung im Ordner `success` einsehen (oder `error` bei fehlerhaften Datensätzen)

Eine Festlegung für „name“ muss auf jeden Fall getroffen werden.

Sollen thumbnails aus Windchill übernommen werden, MUSS der Bezeichnung „thumbnail“ der Wert von „id“ zugeordnet werden[, da die Programmlogik auf den immergleichen Aufbau dieses Wertes zugreift und mit dessen Hilfe die nötige Grafikbezieht].

Parameter werden als „P:Parametername=“ übergeben,

Dimensionen mit „D:Dimensionsname=“

Sprachabhängige Benennungen mit „title_Länderkürzel=“

Neben „name“ sind „path“, „info“ und „objType“ standardmäßig in `GT_Library` enthalten, müssen also nicht als Parameter oder Dimension extra gekennzeichnet werden.

Die zugehörigen Zahlenwerte für „objType“ lassen sich der Beispieldatei entnehmen. Sie sind vom Dateityp des entsprechenden Objektes abhängig

Der Prüfwert für Objektänderungen ist als Parameter zu übergeben. (z.B. `P:CHANGEDATE`)

In der CFG unter „`gtl_dataimporter_change_parameter`“ ist nur der Name (z.B. `CHANGEDATE`) zu hinterlegen. Sobald erstmals so im Objekt angelegt, lässt es sich später bei Updates immer daran prüfen, solange in der CFG kein neuer Prüfparameter hinterlegt wird.

Wird auf diesen Wert verzichtet, werden Updates immer durchgeführt, auch wenn eine ältere Version gleichen Namens benutzt wird, oder sich nichts am ,Objekt geändert hat.

Der Pfad der XSLT-Datei ist unter „`gtl_dataimporter_wc_rest_xslt =`“ zu hinterlegen. Dies kann absolut oder relativ zu `dataImporter.exe` erfolgen

Um zu gewährleisten, dass nur bestimmte Ordner einer Bibliothek ausgelesen werden, wird der gesamte Prozess der XSLT-Datei von einer entsprechenden Anweisung umschlossen.

Bsp.:

```
<xsl:if test="normalize-space(attributes/container) = 'Genius Tools
Library' and normalize-space(attributes/folder)='/Default/test_gle'">
```

...(alle anderen Anweisungen)

```
</xsl:if>
```

Das Beispiel liest nur Objekte aus der Bibliothek GENIUS TOOLS Library (attributes/container) und dem Ordner test_gle (attributes/folder) aus.

Alternativ ist im Beispiel eine auskommentierte Variante enthalten, die gezielt Ordner aus Windchill ausschließt. Diese bietet sich an, wenn mehrere Ordner ausgelesen werden sollen und nur bestimmte nicht und/oder neben den Ordnern noch lose Objekte liegen, die ebenfalls gelesen werden sollen.

```
<xsl:if test="normalize-space(attributes/container) = 'Genius Tools
Library' and normalize-space(attributes/folder)!='Default/config' and
normalize-space(attributes/folder)!='Default/frm_formats' and normalize-
space(attributes/folder)!='Default/gph_groups' and normalize-
space(attributes/folder)!='Default/start_templates'">
```

...(alle anderen Anweisungen)

```
</xsl:if>
```

Das Beispiel liest alle Ordner aus der Bibliothek GENIUS TOOLS Library (attributes/container) außer den Ordnern (attributes/folder) config, frm_formats, gph_groups und start_templates aus.

Denkbar wäre auch eine Kombination aus beiden Anweisungen, gemäß den logischen Regeln des XSLT-Formats.

9.1.5. Programmstart

Siehe Datei 1_Konfiguration_allgemein.docx zur Übergabe von Startargumenten.

DataImporter.exe ausführen

9.2. Datenübertragung via RestApi (Alternative)

Wenn aus irgendeinem Grund keine savedSearch eingerichtet werden kann, gibt es eine Alternative, um dennoch die Rest-Schnittstelle nutzen zu können.

9.2.1. Einrichten der Konfigurationsoptionen

Es werden weitgehend dieselben Konfigurationsoptionen wie für die SavedSearch benötigt.

configOption	Beispielwert
gtl_dataimporter_db_path =	Drive:\Folder_Structure\gt_resource_folder\library\GTL_db.db
gtl_dataimporter_db_thumbnail_path =	Drive:\Folder_Structure\gt_resource_folder\library\GTL_db\
gtl_dataimporter_wc_server =	http://WT-Server:PortNr/Windchill
gtl_dataimporter_wc_login_name =	wcadmin
gtl_dataimporter_wc_login_password =	wcadmin
gtl_dataimporter_wc_rest_xslt =	..\conf\rest.xslt
gtl_dataimporter_change_parameter =	CHANGEDATE

gtl_dataimporter_wc_rest_query =	http://<Windchillserver:Port>/Windchill/servlet/rest/structure/objects?%24filter=(endswith(name%2C%20'prt')%20or%20endswith(name%2C%20'asm')%20%20or%20endswith(name%2C%20'drw')%20or%20endswith(name%2C%20'gph')%20or%20endswith(name%2C%20'mfg')%20or%20endswith(name%2C%20'lay')%20or%20endswith(name%2C%20'frm')%20or%20endswith(name%2C%20'tbl')%20or%20endswith(name%2C%20'sym')%20or%20endswith(name%2C%20'txt')%20or%20endswith(name%2C%20'sec'))%20and%20state%20eq%20'RELEASED'&%24select=CADName%2Ccontainer%2Cfolder%2Cversion%2CMATERIAL%2CDESCRIPTION_1_DE%2CDESCRIPTION_2_DE%2CDESCRIPTION_1_EN%2CDESCRIPTION_2_EN%2CthePersistInfo.modifyStamp&typeId=wt.epm.EPMDocument&queryLimit=
gtl_dataimporter_rest_query_int =	10
gtl_dataimporter_rest_query_days =	1

Unter „gtl_dataimporter_wc_rest_query =“ wird eine URL-eingetragen, die den Pfad zur Rest-API in Windchill enthält an den im Prinzip die SELECT-daten wie bei der SavedSearch angehängt werden.

Gleichbleibend ist immer folgender Teil:

http:// ServerAddress:Port/Windchill/servlet/rest/structure/objects

wobei der jeweilige Servername und Port eingetragen werden muss.

Daran schließt ein Filter an, hinter dem z.B. die zur Übertragung **gewünschten Dateitypen** und deren **Lebenszyklusstatus** eingefügt werden:

?%24filter=

(endswith(name%2C%20'prt')%20or%20endswith(name%2C%20'asm')%20%20or%20endswith(name%2C%20'drw')%20or%20endswith(name%2C%20'gph')%20or%20endswith(name%2C%20'mfg')%20or%20endswith(name%2C%20'lay')%20or%20endswith(name%2C%20'frm')%20or%20endswith(name%2C%20'tbl')%20or%20endswith(name%2C%20'sym')%20or%20endswith(name%2C%20'txt')%20or%20endswith(name%2C%20'sec'))%20and%20state%20eq%20'**RELEASED**'

Dieser wird ermittelt über den Namen des Objekts insbesondere dessen Endung die %-Angaben dazwischen stehen für Zeichen, die so in einer URL nicht auftreten dürfen.

z.B.

- %2C = ,
- %20 = Space
- %24 = \$

→ „(endswith(name%2C%20'prt')“ usw.

Darauf folgt der eigentliche **SELECT**, der enthält, welche **Attribute** der zuvor gefilterten Objekte angezeigt werden sollen:

&%24select=CADName%2Ccontainer%2Cfolder%2Cversion%2CMATERIAL%2CDESCRIPTION_1_DE%2CDESCRIPTION_2_DE%2CDESCRIPTION_1_EN%2CDESCRIPTION_2_EN%2CthePersistInfo.modifyStamp

Am Ende steht jeweils UND-verknüpft die TypenID und eine Angabe, wieviele Datensätze mit der Abfrage geholt werden sollen.

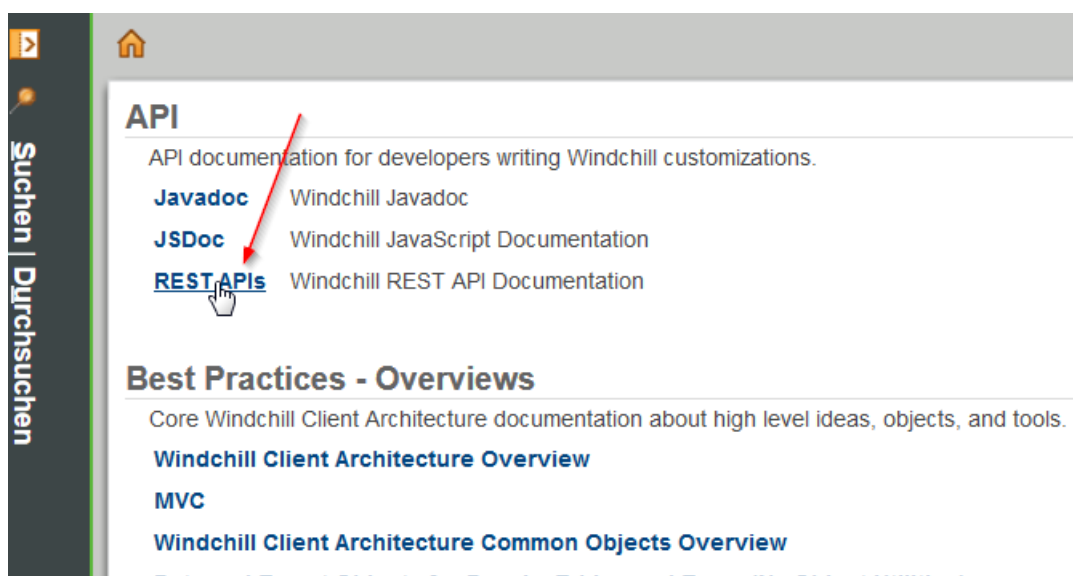
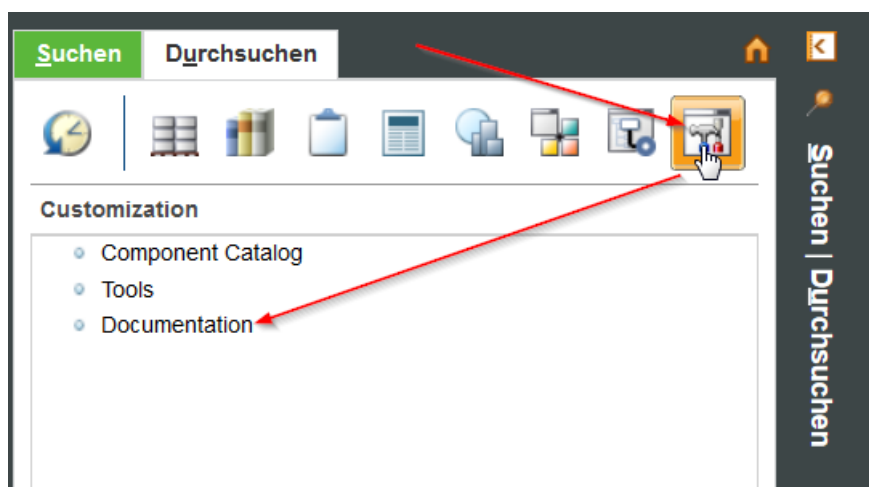
&typeld=wt.epm.EPMDocument&queryLimit=

Wichtig ist, dass das queryLimit= am Ende NICHT mit einem Wert versehen wird. Siehe dazu 1.2.3

Die Abfrage lässt sich entweder nach dem Beispiel der main.cfg aufbauen oder mit der Windchill-eigenen SWAGGER-UI ausgeben.

9.2.2. Swagger UI

Die Swagger UI ist ein Hilfstool, mit dem sich Rest-Abfragen testen lassen. Es findet sich wie folgt:



Die obige URL wird unter dem Punkt Structure Navigation erstellt:

Structure Navigation

Show/Hide

List Operations

Expand Operations

TableView

GET

/structure/objects/{objectId}/designfiles

Get a list of the object's design files' metadata.

Dort klickt man auf /structure/objects

Structure Navigation

Show/Hide

List Operations

Expand Operations

GET	/structure/objects/{objectId}/designfiles	Get a list of the object's design files' metadata.
GET	/structure/objects/{objectId}	Fetch the representation of an object by specifying the Windchill object's OID String.
GET	/structure/objects/{objectId}/ancestors	Get an object's ancestors resource to any number of levels.
GET	/structure/objects/{objectId}/descendants	Get an object's descendants resource to any number of levels.
GET	/structure/objects	Fetch the representation of an object version.
GET	/structure/objects/{objectId}/descendantSummaries	Get resources that summarize how the given resource uses a descendant resource.
GET	/structure/objects/{objectId}/ancestorSummaries	Get resources that summarize how the given resource is used by an ancestor resource.

Und scrollt dann bis zu den Eingabefeldern unter Parameters:

Direkt darüber in der Auswahl kann man das Ausgabeformat json eingeben

Response Content Type


application/json

application/json

application/xml

Parameters

Parameter	Value
\$filter	(endswith(name, 'prt') and

Parameters 				
Parameter	Value	Description	Parameter Type	Data Type
\$filter	<input type="text" value="(required)"/>	<p>An OData filter criteria in the form of "<propertyName> eq '<value>' [and ...]" where <property-name> is either 'number' or 'name'. For example:</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre>\$filter=number eq '0000222341' or name eq 'PTC' or startswith(name, 'PTC') or view eq 'wt.vc.views.View:21281' and state eq 'INWORK' and iterationIdentifier eq '1' and versionIdentifier eq 'A'</pre> </div>	query	string
navigationCriteria	<input type="text"/>	Windchill OID of the navigation criteria or the navigation criteria name. This is used to choose the version of the identified object. If not specified, then the latest version is chosen.	query	string
\$select	<input type="text"/>	An OData select expression that is a comma-separated list of property names.	query	string
typeId	<input type="text" value="(required)"/>	The type on which the query should happen, must be instance of Iterated class e.g. WCTYPE wt.doc.WTDocument or wt.doc.WTDocument.Supports only one type at a time.	query	string
queryLimit	<input type="text"/>	The number of results which the query should return	query	integer

Hier finden sich jetzt die oben aufgezählten Elemente \$filter, \$select, typeId und queryLimit

Die gewünschten Eingaben können hier ohne Sonderzeichen durchgeführt werden.

Für Filter zum Beispiel:

\$filter	<input type="text" value="(endswith(name, 'prt') and state eq 'RELEASES)"/>
-----------------	---

Wobei die Dateitypen ODER-Verknüpft werden und andere Angaben wie Lebenszyklusstatus mit UND(siehe oben)

\$filter	<input type="text" value="ndswith(name, 'prt') or endswith(name, 'asm"/>
-----------------	--

Im Anschluss kann die Selektion der gewünschten Attribute erfolgen. Sie werden einfach kommagetrennt eingefügt. Sie können zum Beispiel aus der main.cfg übernommen und einkopiert werden.

\$select

Schließlich kann unter typeId eine Iterierbar Klasse eingefügt werden, aus der die gewünschten Objekte stammen. wt.epm.EPMDocument bietet sich an.

typeId

Unter queryLimit sollte pro forma ein Wert eingetragen werden, damit es in der URL automatisch mit eingefügt wird. s.u.

queryLimit

Im Anschluss auf den Try it out Button klicken:

queryLimit

Response Messages

HTTP Status Code	Reason	Response
400	If one of the URL or query parameters is not in the correct format.	
404	If the specified objects does not exist.	
500	If an unexpected error occurs.	

[Try it out!](#) [Hide Response](#)

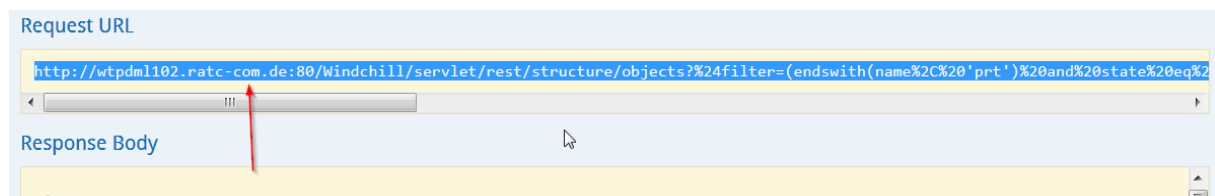
Jetzt beginnt das System die Eingaben zu bearbeiten das kann je nach Datensatzmenge einen Moment dauern.

Wenn man nach unten Scrollt erscheint unter Response Body dann eine Ausgabe

Response Body

```
{
  "items": [
    {
      "id": "OR:wt.epm.EPMDocument:30758899",
      "typeId": "WCTYPE|wt.epm.EPMDocument|de.ratc_com.DefaultEPMDocument",
      "attributes": {
        "container": "Maschinenbau",
        "folder": "/Default/CAD-Daten",
        "version": "A.5",
        "CADName": "200_203_traegerplatte.prt",
        "MATERIAL": "S235JR",
        "thePersistInfo.modifyStamp": "2018-10-18T10:53:17Z"
      }
    },
    {
      "id": "OR:wt.epm.EPMDocument:29758012",
      "typeId": "WCTYPE|wt.epm.EPMDocument|de.ratc_com.DefaultEPMDocument",
      "attributes": {
        "container": "Maschinenbau",
        "folder": "/Default/CAD-Daten",
        "version": "A.5",
        "CADName": "200_203_traegerplatte.prt",
        "MATERIAL": "S235JR",
        "thePersistInfo.modifyStamp": "2018-10-18T10:53:17Z"
      }
    }
  ]
}
```

Sind alle gewünschten Daten enthalten, die man zukünftig verarbeiten möchte, lässt sich jetzt bequem die URL direkt über Response Body unter Request URL kopieren und in die Konfigurationsoptionen eintragen.



Im entsprechenden Editor ist dann darauf zu achten, die Zahl zu queryLimit zu löschen, damit die URL wie folgt aussieht (Ausschnitt):

```
:RIPTION_2_EN%2CthePersistInfo.modifyStamp&typeId=wt.epm.EPMDocument&queryLimit=
```

Damit sind alle notwendigen Einstellungen bei Swagger vorgenommen.

9.2.3. Query Intervall und Zeitraum

Je nachdem, wie oft Daten importiert werden sollen, wird unter

gtl_dataimporter_rest_query_days =

der Zeitraum in Tagen eingetragen, für den vom aktuellen Datum an, Objekte aus Windchill importiert werden sollen.

Bei häufigen Änderungen bietet sich ein kleiner Zeitraum von 1-2 Tagen an, bei gelegentlichen Änderungen von z.B. 7 Tagen.

Wird GTL DI regelmäßig gestartet bietet sich an den Zeitraum mit der Frequenz der Programmausführung zu synchronisieren, also den Zeitraum so groß zu wählen, dass er identisch ist mit dem Zeitraum seit der letzten Durchführung um keine Änderungen zu verpassen.

Bei Verwendung des Changedate-Parameters in den Objekten wird gewährleistet, dass auch bei überlappenden Zeiträumen nichts überschrieben wird, was bereits geändert wurde.

Die Zahl hinter dem = von queryLimit, wird von der Logik GTL Dis automatisch eingefügt. Sie wird unter

`gtl_dataimporter_rest_query_int =`

eingefügt. So wird gewährleistet, dass sicher alle neuen Datensätze erfasst werden, was bei einer statischen Eingabe in der URL eventuell nicht der Fall wäre.

Die Zahl sollte sich an der durchschnittlichen Anzahl zu übertragender Objekte orientieren. Wird GTL DI z.B. täglich ausgeführt und wurden im Schnitt 10 Objekte pro Tag verändert, ist 10 eine sinnvolle Eingabe. sollten größere Mengen erwartet werden z.B. weil die Daten seltener übertragen werden, oder häufig viele kleine Änderungen anfallen, ist die Zahl entsprechend größer zu wählen.

Dies dient vor allem der zeitlichen Optimierung des Datenimportes.

Werden einer oder beiden Konfigurationsoptionen kein Wert oder der Wert 0 zugeordnet, wird stets der komplette Datensatz abgerufen, abgeglichen und verarbeitet, was entsprechend länger dauert, sich allerdings für den ersten Start anbietet, sollen alle Daten nach GTL übertragen werden.

9.2.4. XSLT-Datei

Zur Übertragung wird eine XSLT-Datei benötigt, diese ist identisch mit jener aus der SavedSearch (s.o.)

9.2.5. Programmstart

Siehe Datei 1_Konfiguration_allgemein.docx zur Übergabe von Startargumenten.

DataImporter.exe ausführen

10. Übertragung aus Fremdsystemen

Neben der Übertragung von Daten aus Windchill ist es auch möglich, Daten aus anderen Systemen in GT Library zu übertragen.

Wie diese Dateien erstellt werden hängt vom Ausgangssystem und ist entsprechend der Hilfe des Ausgangssystems zu entnehmen.

GTL_DI bietet zwei Möglichkeiten Fremdsystemdaten zu importieren:

- CSV (mit und ohne Kopfzeile)
- XML

Sämtliche Regeldateien befinden sich als Beispiele/Vorlagen unter

Drive:\Folder_Structure\GT_Library_DataImporter\software\conf

Die einzelnen Konfigurationen werden in den jeweiligen Unterpunkten beschrieben.

Mögliche zu verarbeitende Dateien finden sich unter

Drive:\Folder_Structure\GT_Library_DataImporter\software\PollingExamples

10.1. Übertragung via CSV

Viele Systeme bieten eine Möglichkeit Datensätze von Objekten als CSV-Datei auszugeben.

Je nach System sind dort die einzelnen Objekte zeilenweise aufgeführt.

Die gewünschten Attribute eines Objektes werden gewöhnlich mit Komma oder Semikolon getrennt.

Je nach System enthält eine CSV-Datei alle gewünschten Objekte oder es wird eine Datei pro Objekt angelegt. GTL DI kann alle diese Varianten verarbeiten.

CSV existiert in zwei Varianten.

- Mit Kopfzeile
- Ohne Kopfzeile

Die Kopfzeile führt die Namen der Attribute auf. Die folgenden Zeilen enthalten dann in selber Reihenfolge die Werte.

Existiert keine Kopfzeile, werden nur die Werte der immer gleichen Reihe nach aufgezählt.

In beiden Fällen befindet sich zwischen jedem Attribut bzw. Wert das Trennzeichen, also Komma, Semikolon oder andere.

Im Idealfall werden nur die Werte enthalten sein, die zu übertragen sind. Eventuell werden aber auch z.B. vollständige Datensätze mit Attributen ausgegeben, die nicht in GT Library übertragen werden sollen.

In allen Fällen gewährleistet eine Regeldatei, dass die relevanten Daten übertragen werden.

Werden einzelne Spalten des CSV nicht benötigt, dann können diese in der Regeldatei ausgelassen werden.

Leere Werte werden in der Zeile durch aufeinanderfolgende Kommata markiert. Als Textqualifier, um das Auftreten des Trennzeichens also z.B. , oder ; innerhalb eines Wertes zu markieren, dienen Anführungsstriche. Sie sollten dann den gesamten Wert umschließen zwischen den beiden relevanten Trennzeichen zu den benachbarten Werten.

Die CSV-Dateien müssen zur Übertragung im Ordner polling hinterlegt werden. Sie müssen zwingend die Endung .csv haben.

Die Regeldatei im Ordner conf.

10.1.1.Regeldateien

Regeldateien sind TXT-Dateien die GTL-DI mitteilen, an welcher Stelle der CSV sich die Werte welchen Attributes befinden.

Beispiele für Regeldateien liegen im Ordner conf als

- csv_rule_header (für CSV mit Kopfzeile)
- csv_rule_headerless (für CSV ohne Kopfzeile)

Sie enthalten stets den Namen der gewünschten Attribute getrennt mit = von der entsprechenden Position.

Für CSV mit Kopfzeile sieht das z.B. so aus:

name = Bezeichnung

Das bedeutet, dass der Wert für name sich in jeder Zeile dort befindet, wo das Wort Bezeichnung sich in der Kopfzeile einordnet.

Ohne Kopfzeile wird die Position numerisch angegeben

name = 0

Das bedeutet der Wert für name befindet sich in jeder Zeile an der ersten (informatorisch nullten) Position.

Die Beispiel Regeldateien enthalten die Attributsbezeichnungen für GT Library

u.a. name, path, objType, title_en, title_de und info

zusätzliche Parameter und Dimensionen lassen sich mit P: bzw D: übergeben.

10.1.2. Konfigurationsoptionen CSV

Konfigurationsoption	Beispiel
gtl_dataimporter_csv_rule_file =	..\conf\csv_rule_header.txt
gtl_dataimporter_csv_header_bool =	1
gtl_dataimporter_csv_separator_char =	,
gtl_dataimporter_db_path =	Drive:\Folder_Structure\gt_resource_folder\library\GTL_db.db
gtl_dataimporter_db_thumbnail_path =*	Drive:\Folder_Structure\gt_resource_folder\library\GTL_db\
gtl_dataimporter_change_parameter =*	CHANGEDATE

*optional

10.1.3. CSV mit Kopfzeile

Ein Beispiel für eine CSV-Datei mit Kopfzeile findet sich in PollingExamples unter csvHeaderKommaBild.csv. Das zugehörige Regelfile im Ordner conf heißt csv_rule_header.txt

In der ersten Zeile der CSV befinden sich die Begriffe, wie sie auch in der Regeldatei zu finden sind. Darunter steht pro Zeile ein Objekt dessen Werte jeweils der Kopfzeile folgend angeordnet und dementsprechend in der Regeldatei zugeordnet sind.

z.B.:

name = bezeichnung
path = pfade
objType = objektTypen
thumbnail = bild
status = status
title_en = titel_en
title_de = titel_de
P:MASSE = P1
P:MATERIAL = P2
P:REVISION = P3
D:LAENGE = D1
D:BREITE = D2
D:HOEHE = D3
info = info
P:CHANGEDATE = P4

Und

```
bezeichnung,pfade,objektTypen,status,titel_en,titel_de,P1,P2,P3,D1,D2,D3,info,bild,P4
06screwU.prt,"%GT_RESOURCE_FOLDER\data\GTL_Test_DB\",10,2,06screwU,06SchraubeU,5g,Stahl allg.,V.06,50mm,5mm,8mm,"http://www.lmgtfy.com",,03.10.2018 06:34:03
07screwU.prt,"%GT_RESOURCE_FOLDER\data\GTL_Test_DB\",10,2,07screwU,07SchraubeU,5g,Stahl allg.,V.07,50mm,5mm,8mm,"http://www.lmgtfy.com",/9j/4AAQSkZJRgABAQEASABIAAD/
08screwU.prt,"%GT_RESOURCE_FOLDER\data\GTL_Test_DB\",10,2,08screwU,08SchraubeU,5g,Stahl allg.,V.08,50mm,5mm,8mm,"http://www.lmgtfy.com",,31.09.2018 06:35:03
09screwU.prt,"%GT_RESOURCE_FOLDER\data\GTL_Test_DB\",10,2,09screwU,09SchraubeU,5g,Stahl allg.,V.09,50mm,5mm,8mm,"http://www.lmgtfy.com",/9j/4AAQSkZJRgABAQEASABIAAD/
10screwU.prt,"%GT_RESOURCE_FOLDER\data\GTL_Test_DB\",10,2,10screwU,10SchraubeU,5g,Stahl allg.,V.10,50mm,5mm,8mm,"http://www.lmgtfy.com",/9j/4AAQSkZJRgABAQEASABIAAD/
11screwU.prt,"%GT_RESOURCE_FOLDER\data\GTL_Test_DB\",10,2,10screwU,10SchraubeU,5g,Stahl allg.,V.10,50mm,5mm,8mm,"http://www.lmgtfy.com",/9j/4AAQSkZJRgABAQEASABIAAD/
13screwU.prt,"%GT_RESOURCE_FOLDER\data\GTL_Test_DB\",10,2,10screwU,10SchraubeU,5g,Stahl allg.,V.10,50mm,5mm,8mm,"http://www.lmgtfy.com",/9j/4AAQSkZJRgABAQEASABIAAD/
14screwU.prt,"%GT_RESOURCE_FOLDER\data\GTL_Test_DB\",10,2,10screwU,10SchraubeU,5g,Stahl allg.,V.10,50mm,5mm,8mm,"http://www.lmgtfy.com",/9j/4AAQSkZJRgABAQEASABIAAD/
```

10.1.4.CSV ohne Kopfzeile

```
name = 0
path = 1
objType = 2
status = 3
title_en = 4
title_de = 5
P:MASSE = 6
P:MATERIAL = 7
P:REVISION = 8
D:LAENGE = 9
D:BREITE = 10
D:HOEHE = 11
info = 12
thumbnail = 13
P:CHANGEDATE = 14
```

```
ohne01screwU.prt,"%GT_RESOURCE_FOLDER\data\GTL_Test_DB\",10,2,01screwU,01SchraubeU,5g,Stahl allg.,V.01,50mm,5mm,8mm,"http://www.l
ohne02screwU.prt,"%GT_RESOURCE_FOLDER\data\GTL_Test_DB\",10,2,02screwU,02SchraubeU,5g,Stahl allg.,V.02,50mm,5mm,8mm,"http://www.l
ohne03screwU.prt,"%GT_RESOURCE_FOLDER\data\GTL_Test_DB\",10,2,03screwU,02SchraubeU,5g,Stahl allg.,V.03,50mm,5mm,8mm,"http://www.l
ohne04screwU.prt,"%GT_RESOURCE_FOLDER\data\GTL_Test_DB\",10,2,04screwU,04SchraubeU,5g,Stahl allg.,V.04,50mm,5mm,8mm,"http://www.l
ohne05screwU.prt,"%GT_RESOURCE_FOLDER\data\GTL_Test_DB\",10,2,05screwU,02SchraubeU,5g,Stahl allg.,V.05,50mm,5mm,8mm,"http://www.l
ohne06screwU.prt,"%GT_RESOURCE_FOLDER\data\GTL_Test_DB\",10,2,05screwU,02SchraubeU,5g,Stahl allg.,V.05,50mm,5mm,8mm,"http://www.l
```

2.2.1 Allgemein XML-Files aus Fremdsystemen

Viele Systeme bieten eine Möglichkeit Datensätze von Objekten als XML-Datei auszugeben.

Diese lassen sich durch eine entsprechende XSLT-Datei für GT Library DataImporter nutzbar machen.

Die genaue Struktur einer solchen XML-Datei hängt von der entsprechenden Quelle ab.

Im Ordner PollingExamples findet sich exemplarisch eine Beispiel XML-Datei (xmlBsp) und im Ordner Conf eine Beispiel XSLT (xml-rule), die darauf abgestimmt ist.

2.2.2 XML-Konfigurationsoptionen

Konfigurationsoption	Beispiel
gtl_dataimporter_xml_rule_file =	..\conf\xml_rule.xslt
gtl_dataimporter_db_path =	Drive:\Folder_Structure\gt_resource_folder\library\GTL_db.db
gtl_dataimporter_db_thumbnail_path =*	Drive:\Folder_Structure\gt_resource_folder\library\GTL_db\
gtl_dataimporter_change_parameter =*	CHANGEDATE

*optional

2.2.3 XML-Files

XML-files sind sehr frei und individuell gestaltbar. Deswegen lässt sich kaum eine allgemeine Erklärung dazu liefern. Für generelle Informationen zum Thema bietet sich die Seite <https://wiki.selfhtml.org> an.

Grundsätzlich gilt, dass die enthaltenen Objektdaten Knoten bzw. Unterknoten zu den entsprechenden Attributen beinhalten sollten, die die entsprechenden Werte enthalten.

Im Idealfall sind Knoten für den Objektnamen, den Pfad, den Objekttypen, Status und allgemeine Informationen enthalten. Weitere Attribute lassen sich im Rahmen von Parametern und Dimensionen übertragen. Für verschiedensprachige Benennungen sind Knoten die title und Länderkürzel übergeben sinnvoll.

Die genaue Knotenstruktur ist der Dokumentation der ausgebenden Applikation zu entnehmen.

Thumbnails lassen sich als base64 kodierte Werte übergeben.

2.2.4 XSLT-Files

Alle Knoten müssen in der XSLT eine Zuweisung nach dem Muster „Attributname=“ erfahren, dem der Wert eines oder mehrere adäquate Knoten zugewiesen wird.

Alle Attribute müssen am Ende die Anweisung

```
<xsl:text>&#xa;</xsl:text>
```

erhalten.

Werden mehrere Objekte innerhalb einer Datei erwartet, sollte die XSLT-Datei eine for-each-Anweisung um die entsprechenden Knoten herum enthalten. Z.B.:

```
<xsl:for-each select="root/element/items"> </xsl:for-each>
```

Die letzte Anweisung für ein Objekt sollte immer

```
<xsl:text>-&#xa;</xsl:text>
```

sein, damit DataImporter bei der späteren Auswertung das Ende eines Objektes erkennt.

Wie auch immer die Knoten im XML benannt sind, die XSLT-Datei bereitet diese für DataImporter auf.

In der Datei xml-rule ist zu sehen, wie Knoten aufgerufen werden, und deren Inhalte einem GT Library konformen Label zugewiesen werden. Beispiele:

```
<xsl:if test="name and normalize-space(name)!=''">
  <xsl:text>name=</xsl:text>
  <xsl:value-of select="normalize-space(name)"/>
  <xsl:text>&#xa;</xsl:text>
</xsl:if>
```

Das obige Beispiel prüft auf einen Knoten mit der Bezeichnung „name“ und weist dessen Value dem Begriff „name=“ zu. Im Anschluss wird eine Enteranweisung angefügt.
Das = und die Enter-Anweisung sind obligatorisch, damit DataImporter die ausgelesenen Values richtig zuordnen kann und erkennt, wo ein Objekt zu Ende ist und das nächste anfängt.

```
<xsl:if test="P[@key='MASSE'] and normalize-space(P[@key='MASSE'])!='' ">
    <xsl:text>P:MASSE=</xsl:text>
    <xsl:value-of select="normalize-space(P[@key='MASSE'])"/>
    <xsl:text>&#xa;</xsl:text>
</xsl:if>
```

Das obige Beispiel liest einen Parameter aus und prüft, was im entsprechenden XML-File als Key innerhalb der Knotenbezeichnung ausgewiesen ist. Hinter der Beschriftung „P:MASSE=“ wird dann der korrespondierende Wert aus dem XML-File eingefügt. Am Ende wird ein Enterzeichen eingefügt.

Der Doppelpunkt zwischen P und Bezeichnung ist obligatorisch, soll DataImporter das Konstrukt als Parameter für Library aufarbeiten. Bei Dimensionen wäre es D: Die Bezeichnung im Beispiel: MASSE muss nicht der Beschriftung des Knotens entsprechen, solange die Zuordnung eindeutig ist.

So könnte der Knoten der XML-Datei auch „<P key = "gewicht"></P>“ heißen, solange dies in der XSLT Datei entsprechend ausgewiesen ist.

```
<xsl:if test="title_en and normalize-space(title_en)!='' ">
    <xsl:text>title_en=</xsl:text>
    <xsl:value-of select="normalize-space(title_en)"/>
    <xsl:text>&#xa;</xsl:text>
</xsl:if>
```

Sprachspezifische Bezeichnungen werden durch den Text: „title_“ eingeleitet und bekommen das entsprechende Länderkürzel de, en oder ähnliches zugewiesen. Dem folgt ein = und dahinter wird der Text der zugewiesenen Knoten angefügt.

Auch hier muss wieder ein Enter eingefügt werden im Anschluss.

11. Andere Dateiformate

Sollten andere Dateiformate, z.B. JSON, vorliegen, die ihren Ursprung nicht in Windchill haben, wird angeraten, diese in XML mit XSLT zu transformieren, bevor diese automatisiert mit DataImporter ausgewertet werden können.