

PLM-EASY FÜR TEAMCENTER



ADMIN TOOLS

Verfasser: Siemens PLM Software
Version: 10.1.2
Datum: September 2014

Copyright

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung die nicht ausdrücklich vom Urhebergesetz zugelassen ist, bedarf der vorherigen Zustimmung der Siemens Industry Software GmbH & Co. KG. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

© by Siemens Industry Software GmbH & Co. KG

1. INHALTSVERZEICHNIS

<i>PLM-Easy für Teamcenter</i>	1
1. <i>Inhaltsverzeichnis</i>	2
2. <i>Änderungslegende</i>	4
<i>Info – Funktionen</i>	5
3. <i>Info - tcpb_backup_dataset_info</i>	5
4. <i>Info - tcpb_find_datasets</i>	8
5. <i>Info - tcpb_list_from_query</i>	11
6. <i>Info - tcpb_volume_report</i>	16
<i>Datenpflege Funktionen</i>	19
7. <i>Maintenance - tcpb_attach_file_to_job</i>	19
8. <i>Maintenance - tcpb_batch_rename</i>	20
9. <i>Maintenance - tcpb_change_ownership</i>	24
10. <i>Maintenance - tcpb_change_volume</i>	27
11. <i>Maintenance - tcpb_checkin_by_config</i>	33
12. <i>Maintenance - tcpb_init_seed_counter</i>	38
13. <i>Maintenance - tcpb_purge_dataset</i>	39
14. <i>Maintenance - tcpb_set_attr_value</i>	41
<i>Data-Exchange Funktionen</i>	42
15. <i>Exchange - tcpb_data_import</i>	42
16. <i>Exchange - tcpb_export_file</i>	42
17. <i>Exchange - tcpb_export_view</i>	46

ADMIN TOOLS

18. Exchange - tcpb_import_file	47
19. Exchange - tcpb_sync_cadattributes	49
CreteImageServer Funktionen	51
20. CRI - tcpb_count_datasets	51
21. CRI - tcpb_ideas_user_attribute	51
22. CRI - tcpb_job_util	51
23. CRI - tcpb_set_decision	52

2. ÄNDERUNGSLEGENDE

Version	Datum	Inhalt, Änderung
1.0	Dezember 2013	Release für TC10.1
1.1	Juli 2014	tcpb_backup_dataset_info geändert
1.2	September 2014	tcpb_batch_rename erweitert

INFO – FUNKTIONEN

3. INFO - TCPB_BACKUP_DATASET_INFO

3.1. Beschreibung

Dieses Programm erstellt eine Liste von Datasets und den dazugehörigen Dateien in den TC Volumes. Hiermit kann man z.B. gezielt Dateien in einem Volumebackup identifizieren.

3.2. Aufruf

```
tcpb_backup_dataset_info [-u=<user> -p=<passwd> -g=<group>] [-ab=<date>] [-bis=<date>] [-incl_noref] [-h|-help]
```

3.3. Argumente

-u	= Login User
-p	= Login Passwort
-g	= Login Gruppe
-ab	= Modifikationsdatum ab dem Datensätze gelistet werden sollen
-bis	= Modifikationsdatum bis zu dem Datensätze gelistet werden sollen
-incl_noref	= Listet auch Datensätze ohne Referenzen, d.h. Dateien in den Volumes. Default: nur Datensätze mit referenzierten Dateien listen.
-h -help	= Ausgabe der Hilfe

3.4. Umgebung

Zum Start des Programms wird eine TC-Command Shell benötigt.

Siehe auch Teamcenter Help ⇒ Configuring Teamcenter ⇒ Preferences and Environment Variables Reference.

3.5. Dateien

Es wird eine Ausgabedatei „backup_dataset_info_<Datum>.log“ erzeugt. Im Falle einer vollständigen Liste lautet der Name der Datei „backup_dataset_info_all.log“. Ansonsten werden die Standard TC Log files angelegt, siehe auch Teamcenter Help ⇒ Administering Teamcenter ⇒ Utilities Reference ⇒ Getting started ⇒ Log files.

3.6. Zu Beachten

Die Parameter -u, -p und -g müssen immer zusammen oder überhaupt nicht (autologin) angegeben werden.

Dieses Programm muss unter einem Benutzer gestartet werden, der auf alle Datensätze der Datenbank Leserechte besitzt.

ADMIN TOOLS

Das Zeitformat des Parameters –ab lautet: TT.MM.YYYY.HH.MM und muss immer vollständig spezifiziert werden.

TT = Tag
MM = Monat
YYYY = Jahr
HH = Stunde
MM = Minute

Folgende Relationen werden per default im Programm benutzt (falls die TC - Preference **TCPB_backup_dataset_info_relations** nicht existiert), um die Id Information der das Dataset referenzierenden Revision zu finden:

IMAN_specification
IMAN_manifestation
IMAN_Rendering
IMAN_UG_altrep

Sollte dies nicht ausreichen, können die Relationen per TC-Preference **TCPB_backup_dataset_info_relations** (string array) spezieller definiert werden.

3.7. Beispiele

3.7.1. Beispiel 1: Liste alle Datensätze

Kommando

```
tcpb_backup_dataset_info -u=infodba -p=<password> -g=dba
```

Ausgabe (STDOUT):

```
Backup dataset info v1.2
Alle Datasets ab Datum suchen und Info raus schreiben ....
~~~~~

MA 2010 y 2 m 1 d 12 h 30 min
1.) DatasetName=Schaft02-A     Typ=UGMASTER ...
2.) DatasetName=UGMASTER ProductVision data     Typ=DirectModel ...
3.) DatasetName=UGMASTER ProductVision data     Typ=DirectModel ...
4.) DatasetName=spec     Typ=UGPART ...
...
```

Ausgabedatei: backup_dataset_info_all.log

```
Alle Datasets ab Datum suchen und Info raus schreiben ....
~~~~~

MA 2010 y 2 m 1 d 12 h 30 min
gefundene Anzahl =26
```

ADMIN TOOLS

```
1.) Id/Rev.=Schafft02/A      Datasetname=[Schafft02-A]
    Typ=UGMASTER            Modifiziert=14-Feb-2010 15:49
    File=[d:\volume01\training_3d57d343\schaft02_a_ugp_3e4d01f9.prt]
2.) Id/Rev.=Schafft01/001    Datasetname=[UGMASTER ProductVision data] Typ=DirectModel   Modifiziert=14- Feb-2010
15:48   File=[d:\volume01\training_3d57d343\none_jt__3e4d01ca.jt]
3.) Id/Rev.=Schafft02/A      Datasetname=[UGMASTER ProductVision data] Typ=DirectModel   Modifiziert=14-Feb-2010
15:49   File=[d:\volume01\training_3d57d343\none_jt__3e4d01fa.jt]
4.) Id/Rev.=Schafft02/A      Datasetname=[spec] Typ=UGPART   Modifiziert=14-Feb-2010 15:49
    File=[d:\volume01\training_3d57d343\spec_ugpart_3e4d0200.prt]
...
```

3.7.2.Beispiel 2: Liste alle Datensätze, welche nach 12:30 Uhr am 01.02.2010 bis heute geändert wurdenKommando

```
tcpb_backup_dataset_info -u=infodba -p=infodba -g=dba -ab=01.02.2010.12.30
```

Ausgabedatei: backup_dataset_info_01.02.2010.12.30.log

4. INFO - TCPB_FIND_DATASETS

4.1. Beschreibung

Das Programm sucht Datensätze nach vorgegebenen Kriterien und speichert diese in einer ASCII Datei. Diese Ausgabedatei kann dann z. B. zu gezielten Refile Aktionen verwendet werden.

4.2. Aufruf

```
tcpb_find_datasets [-u=] [-p=] [-g=] -dstype=<dataset type> -relation=<relation type>
                  [-format=(CSV|CSVFULL|REFILE)]
                  [-owner=<user_id>] [-group=<groupname>] [-status=(<status type>|all|none)]
                  [-modifiedbefore=(TT.MM.YYYY-HH:MM)] -file=<outputfile> -log=<logfile> [-help]
```

4.3. Argumente

-u	= Login User
-p	= Login Password
-g	= Login Group
-dstype	= Datensatz Typ
-relation	= Datensatz Relation zur zugehörigen ItemRevision
-format	= Ausgabeformat: CSV Datei (ItemId;RevID;Datasetname) oder Refile-Input-Datei
-owner	= Eigentümer
-group	= Eigentümer-Gruppe
-status	= Objektstatus
-modifiedbefore	= Ausschluss von aktuellen Datensätzen (z.B. nach dem Upgrade schon gespeichert)
-file	= Ausgabedatei
-log	= Logdatei
-help	= Ausgabe der Hilfe

4.4. Zu Beachten

Defaults:

- format = REFILE
- owner = all owners
- group = all groups
- status = all status

Bei format=REFILE, sind nur die Relationen **IMAN_specification**, **IMAN_UG_altrep** und die Datensatztypen **UGMASTER**, **UGPART**, **UGALTREP** erlaubt.

4.5. Umgebung

Zum Start des Programms wird eine TC-Command Shell benötigt.

Siehe auch Teamcenter Help ⇒ Configuring Teamcenter ⇒ Preferences and Environment Variables Reference.

4.6. Dateien

Es werden die Standard TC Log files angelegt, siehe auch Teamcenter Help ⇒ Administering Teamcenter ⇒ Utilities Reference ⇒ Getting started ⇒ Log files.

4.7. Beispiel 1

```
tcpb_find_datasets -u=infodba -p=<password> -g=dba -dstype=UGMASTER -relation=IMAN_specification -format=REFILE  
-owner=edsplm1 -file=ugmaster_list.txt -log= ugmaster_list.log
```

Ausgabe:

```
tcpb_find_datasets V9.1.2.1  
Logfile is set to: ugmaster_list.log
```

Ausgabedatei: ugmaster_list.txt

```
@DB@UG_DIN_A0@A  
@DB@UG_DIN_A1@A  
@DB@UG_DIN_A3@A  
@DB@UG_DIN_A2@A  
@DB@cap_screw_cs_inch@A  
@DB@cap_screw_cb_inch@A  
@DB@standard_thread_metric@A  
@DB@drill_inch@A  
@DB@cap_screw_cs_metric@A
```

Logdatei (evtl. Fehlerprotokoll): ugmaster_list.log

```
Logfile: tcpb_find_datasets V9.1.2.1
```

```
DATASET TYPE: UGMASTER  
DATASET RELATION: IMAN_specification  
DATASET OWNING USER: edsplm1  
DATASET OWNING GROUP: (null)  
DATASET STATUS: (null)  
OUTPUT FORMAT: REFILE  
OUTPUT: ugmaster_list.txt  
LOGFILE: ugmaster_list.log
```

4.8. Beispiel 2

```
tcpb_find_datasets -u=infodba -p=<password> -g=dba -dstype=UGMASTER -relation=IMAN_specification -format=CSV  
-owner=edsplm1 -file=ugmaster_list.txt -log= ugmaster_list.log
```

Ausgabe:

```
tcpb_find_datasets V9.1.2.1  
Logfile is set to: ugmaster_list.log
```

Ausgabedatei: ugmaster_list.txt

```
UG_DIN_A0;A;UG_DIN_A0-A  
UG_DIN_A1;A;UG_DIN_A1-A  
UG_DIN_A3;A;UG_DIN_A3-A  
UG_DIN_A2;A;UG_DIN_A2-A  
cap_screw_cs_inch;A;cap_screw_cs_inch  
cap_screw_cb_inch;A;cap_screw_cb_inch  
standard_thread_metric;A;standard_thread_metric  
drill_inch;A;drill_inch  
cap_screw_cs_metric;A;cap_screw_cs_metric
```

Logdatei (evtl. Fehlerprotokoll): ugmaster_list.log

Logfile: tcpb_find_datasets V9.1.2.1

```
DATASET TYPE: UGMASTER  
DATASET RELATION: IMAN_specification  
DATASET OWNING USER: edsplm1  
DATASET OWNING GROUP: (null)  
DATASET STATUS: (null)  
OUTPUT FORMAT: CSV  
OUTPUT: ugmaster_list.txt  
LOGFILE: ugmaster_list.log
```

5. INFO - TCPB_LIST_FROM_QUERY

5.1. Beschreibung

Das Programm führt eine Suchabfrage (Search-Query) durch, die entweder zu den Standardsuchabfragen gehört oder vorher interaktiv definiert worden sein muss. Die gefundenen Objekte werden in eine Datei geschrieben. Jede Zeile der Datei entspricht einem Objekt. Der Aufbau der Zeilen ist abhängig von der Suchklasse der verwendeten Query. In jedem Fall ist der erste Eintrag der Tag des Objektes, also der SID-interne Kenner. Durch das Zeichen | getrennt folgen dann weitere Einträge wie z.B. der „Type“ und der Name des Objektes. Mit Hilfe des Tags ist es möglich, die erzeugte Datei wiederum als Eingangsdatei für andere ITK-Programme zu verwenden und so die gefundenen Objekte zum Beispiel mit einem Status zu versehen oder sie in ein Ziel-Volume zu verschieben.

5.2. Aufruf

```
tcpb_list_from_query.exe [-u=<user>] [-p=<password>] [-g=<group>] -query=<query-name> [-param]=<parameters of the query> [-output]=<output file> [-log]=<log file> [-h|elp]]
```

5.3. Argumente

-u	= Login User
-p	= Login Password
-g	= Gruppe
-query	= query-name
-param	= parameters of the query, separated by ";". default: *;
-output	= output file, default: tcpb_list_from_query.txt
-log	= log file, default: tcpb_list_from_query.log
-h -help	Ausgabe der Hilfe

Nur der Parameter -query ist Pflichtparameter. Bei Leerstellen im Namen der Query diesen maskieren. Beispiel: "Item Revision..."

Bei der durch Semikolon getrennten Liste der Suchabfragen-Parameter (-param) kann \$DEFAULT eingetragen werden. Sofern die Suchabfrage hier einen Default-Wert hat, wird dieser übernommen.

5.4. Umgebung

Zum Start des Programms wird eine TC-Command Shell benötigt.

Siehe auch Teamcenter Help → Configuring Teamcenter → Preferences and Environment Variables Reference.

5.5. Dateien

Keine

5.6. Zu Beachten

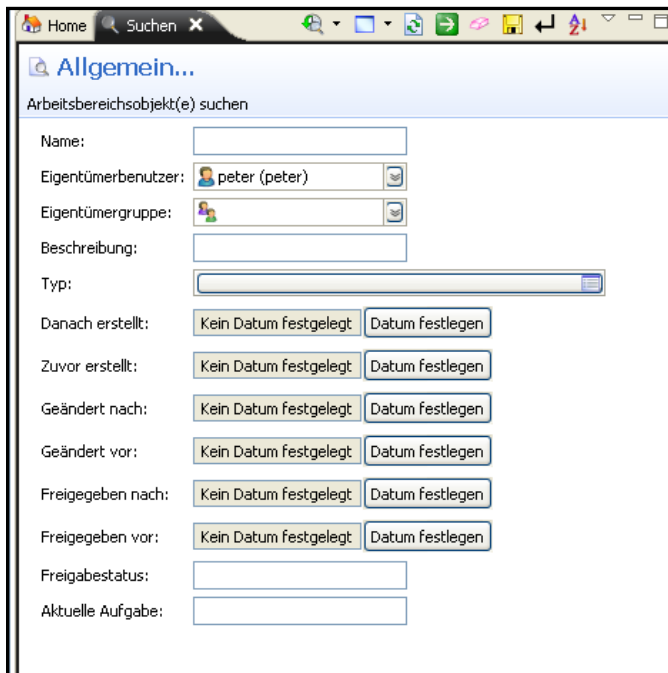
Siehe Anmerkungen in den jeweiligen Beispielen.

5.7. Beispiele

5.7.1. Beispiel 1:

Vor dem Aufruf TC-Umgebung setzen! (folgender Aufruf in einer Zeile)

```
tcpb_list_from_query.exe -u=infodba -p=infodba -g=dba -query=General... -param=;;;peter;;;;;;;; -output=query_out2.txt  
-log=logfile.txt
```



Obiger Aufruf des Programms entspricht einer Abfrage, wie sie im Bild dargestellt ist.

Die Parameter (*-param=;;;peter;;;;;;;;*) müssen exakt in der Position stehen, in der sie in der interaktiven Suchabfrage stünden, wobei der erste unmittelbar nach dem Gleichheitszeichen vor dem ersten Semikolon einzutragen ist. Da der User-Eintrag im Beispiel an **viertes** Stelle im interaktiven Menü zu tätigen ist, befinden sich im Programmaufruf deshalb **drei** Semikolons vor dem Eintrag. Die nachfolgenden Semikolons können entfallen, wenn kein weiterer Eintrag erfolgen soll.

VOREINSTELLUNGEN:

Die Einträge *-u=...* *-p=...* *-g=...* können entfallen, wenn Autologin gesetzt ist.

Wird *-param* nicht eingegeben, wird automatisch „*“ verwendet. Die Suchabfrage findet dann alle Objekte in der Datenbank.

-output und *-log* werden wie oben beschrieben gesetzt, falls ihr Wert nicht beim Aufruf des Programms explizit gesetzt wird. Sie werden in jedem Fall in die Verzeichnisse platziert, von denen aus der Aufruf erfolgt.

5.7.2. Beispiel 2:

```
tcpb_list_from_query.exe -query=General... -param=;;;peter
```

Dieser Aufruf entspricht vom Ergebnis her exakt dem des ersten einführenden Beispiels. Die Suchklasse ist "workspaceobject". In der Datei *tcpb_list_from_query.txt* finden sich folgende Einträge (4. Fall „Sonst“):

```
A1BBYSm$F17FUCAAAAAAAAAAAAAA|ItemRevision|r3test|peter
A1EBYSm$F17FUCAAAAAAAAAAAAAA|view BOMView Revision|r3/b-view|peter
A1MBX0DPFI7FUCAAAAAAAAAAAAAA|UGPartBoundingBoxForm|000018/A|peter
A5KBYSm$F17FUCAAAAAAAAAAAAAA|ItemRevision Master|r3/b|peter
A5MBX0DPFI7FUCAAAAAAAAAAAAAA|UGPartAttributesForm|000018/A|peter
A9JBYSm$F17FUCAAAAAAAAAAAAAA|ItemRevision|r2|peter
A9PBYSm$F17FUCAAAAAAAAAAAAAA|ItemRevision Master|r2/b|peter
ABABIHBIFI7FUCAAAAAAAAAAAAAA|Item|s1|peter
ABABYilwFI7FUCAAAAAAAAAAAAAA|Job|r1/a|peter
....
....
```

5.7.3. Beispiel 3:

```
tcpb_list_from_query.exe -u=infodba -p=infodba -g=dba -query="Item..." -param=;;;;;Item; -output=query_out.txt
```

Item... hat die Suchklasse "Item", entspricht also dem ersten Fall. In der Datei *query_out.txt* sind folgende Einträge:

```
A5y89ZhsFI7FUCAAAAAAAAAAAAAA|Item|cap_screw_cb_inch|cap_screw_cb_inch|infodba
ABABIHBIFI7FUCAAAAAAAAAAAAAA|Item|s1|s1|peter
ABEBTmurFI7FUCAAAAAAAAAAAAAA|Item|000010|g|infodba
ABMBm1EBFI7FUCAAAAAAAAAAAAAA|Item|teil2|teil2|infodba
ABNBWjldFI7FUCAAAAAAAAAAAAAA|Item|neueForm?|neueForm?|infodba
AFNBIHBIFI7FUCAAAAAAAAAAAAAA|Item|s2|s2|peter
AJBBm1EBFI7FUCAAAAAAAAAAAAAA|Item|assem|assem|infodba
AJEBYCVEFI7FUCAAAAAAAAAAAAAA|Item|assemzumpasten|a|infodba
AJFBD_JQFI7FUCAAAAAAAAAAAAAA|Item|000007|gg|infodba
```

5.7.4.Beispiel 4:

```
tcpb_list_from_query.exe -u=infodba -p=infodba -g=dba -query="Item..." -param=;;; $DEFAULT;;; Document -output=query_out.txt
```

erzeugt folgende Fehlermeldung:

Can't execute query! returncode=3006

Das Log file ist default auf *tcpb_list_from_query.log* gesetzt:

Log file: tcpb_list_from_query v0.0

QUERY: Item...

PARAMETER: ;;; \$DEFAULT;;; Document

OUTPUT: query_out.txt

Query values: new_entries: <Alias Rev ID> new_values: <\$DEFAULT>;

Query values: new_entries: <Type> new_values: <Document>;

Can't execute query! returncode=3006

Man erkennt, dass Alias Rev ID den Default-Wert übernehmen sollte. Hierfür ist aber gar keiner gesetzt.

5.7.5.Beispiel 5:

```
tcpb_list_from_query.exe -u=infodba -p=infodba -g=dba -query="Item..." -param=;;;;; Document;$DEFAULT;$DEFAULT -
output=query_out.txt
```

Hier wird für den owning-user und die owning-group der Default übernommen. Das Ergebnis beinhaltet diesmal nur die Objekte der Klasse Item vom Typ Document:

AFKBKfanFI7FUCAAAAAAAAAAAAAAAA|Document|destr|destr|infodba

AREBKfanFI7FUCAAAAAAAAAAAAAAAA|Document|4712|4712|infodba

Log file: tcpb_list_from_query v0.0

QUERY: Item...

PARAMETER: ;;;;;; Document;\$DEFAULT;\$DEFAULT

OUTPUT: query_out.txt

Query values: new_entries: <Type> new_values: <Document>;

Query values: new_entries: <Owning User> new_values: <infodba>;

Query values: new_entries: <Owning Group> new_values: <dba>;

ADMIN TOOLS

Found <2> objects
Query class: [Item]
returncode=0

OUTPUT:

Die Formatierung des Output-Files erfolgt in Abhängigkeit von der Suchklasse der Query. Es werden folgende Fälle unterschieden:

Format der Zeilen:

- Die Suchklasse ist Item:
tag|Type|ItemID|Name|OwningUser
- Die Suchklasse ist ItemRevision:
tag|Type|ItemID|RevisionID|Name|OwningUser
- Die Suchklasse ist Dataset:
tag|Type|Name|Dataset-Version|OwningUser
- Sonst:
tag|Type|Name|OwningUser

6. INFO - TCPB_VOLUME_REPORT

6.1. Beschreibung

Tool zum Erstellen eines Reports über den von einzelnen Anwendern bzw. von einzelnen Gruppen belegten Plattenplatz.

6.2. Aufruf

```
tcpb_volume_report -u= -p= -g= -owner=<user id> -ow_group=<group_name> [-out_group=<filename>]  
[-out_user=<filename>] [-out_detail=<filename>] [-tagfile=<filename>] [-bypass=YES]  
[-mode=<full|taglist|rep_from_list>] [-max_entries=<integer>] [-output_mode=[create|append]]
```

6.3. Argumente

-u	= Login User
-p	= Login Passwort
-g	= Login Gruppe
-owner	= User, für den ein Report erstellt werden soll
-ow_group	= Gruppe, für die ein Report erstellt werden soll
-out_user	= Ausgabedatei für Dateien pro Eigentümer Report (default: output_user.txt)
-out_group	= Ausgabedatei für Dateien pro Gruppe Report (default: output_group.txt)
-out_detail	= Ausgabedatei für detaillierte Informationen (default: output_detail.txt)
-tagfile	= Taglisten Datei
-bypass	= Programmabarbeitung im bypass-mode(YES), (default: NO)
-mode	= <full taglist rep_from_list> (default: full) full = Vollständiger Report taglist = Ausgabe einer Liste von Datensatztags rep_from_list = detailed Report aus einer Eingabeliste von Datensatztags
-max_entries	= Maximale Anzahl von Tags in der Tagliste (default: 1000)
-output_mode	= create = Ausgabedateien neu erstellen append = Report an bestehende Dateien anhängen
-h -help	Ausgabe der Hilfe

6.4. Umgebung

Zum Start des Programms wird eine TC-Command Shell benötigt. Siehe auch Teamcenter Help ⇨ Configuring Teamcenter ⇨ Preferences and Environment Variables Reference.

6.5. Dateien

Es werden drei Ausgabedateien angelegt:

output_user.txt	Report des vom Anwender belegten Speicherplatz (Angabe in Byte)
output_group.txt	Report des von der Gruppe belegten Speicherplatz (Angabe in Byte)
output_detail.txt	Report über alle Datensätze, mit Angabe von Ownerinformationen, Dateigröße und Dateipfad.

Weiterhin werden die Standard TC Log files angelegt, siehe auch Teamcenter Help ⇒ Administering Teamcenter ⇒ Utilities Reference ⇒ Getting started ⇒ Log files.

6.6. Zu Beachten

Die Parameter -u, -p und -g müssen immer zusammen oder überhaupt nicht (autologin) angegeben werden.

Der Parameter -bypass=YES bedingt, dass der Anwender Bypass-Rechte besitzen muss (Mitglied der DBA-Gruppe)

Ist die Anzahl der zu bearbeitenden Datensätze sehr hoch, dann sollte man mit dem Taglistenverfahren arbeiten (siehe Beispiel 2)

6.7. Beispiele

6.7.1. Beispiel 1: Einfacher Report über alle Informationen

Kommando

```
tcpb_volume_report -u=infodba -p=<password> -g=dba
```

Ausgabedatei: output_user.txt

```
owner;total_size  
infodba;29506221  
tiffplot;237129
```

Ausgabedatei: output_group.txt

```
owning_group;total_size  
Konstruktion1;3494444  
dba;29939804
```

Ausgabedatei: output_detail.txt

```
owner;owning_group;volume_name;file_size;object_name;revision_number;total_number_of_revisions;number_of_references;file_path  
infodba;dba;vol01;41651;09001;2;2;4;i:\iman\im7a\volumes\vol01\dba_3c7f7565\p000001_00_ugp_3ce5110a.prt;  
infodba;dba;vol01;64789;09010;2;3;3;i:\iman\im7a\volumes\vol01\dba_3c7f7565\15790_00_ugpart_3cfc593d.prt;
```

ADMIN TOOLS

6.7.2. Beispiel 2: Detailed Report über Tagliste (Teilelisten von 1000 Zeilen)Schritt1:Kommando:

```
tcpb_volume_report -u=infodba -p=<password> -g=dba -mode=taglist -taglist=liste -max_entries=1000
```

Ausgabedateien:

taglist.bat	automatisch generiertes Batch Script aus Schritt1
taglist000000001.txt	Tagliste für Dataset 1-1000
taglist000000002.txt	Tagliste für Dataset 1001-2000
.....	

Schritt2:Kommando:

```
taglist.bat
```

Ausgabedateien:

out_taglist000001.txt	Detailed Report für Datensätze aus Tagliste taglist000000001.txt
out_taglist000002.txt	Detailed Report für Datensätze aus Tagliste taglist000000002.txt
.....	

DATENPFLEGE FUNKTIONEN

7. MAINTENANCE - TCPB_ATTACH_FILE_TO_JOB

7.1. Beschreibung

Dieses Programm dient dazu, eine Logdatei als Referenz an einen Job anzuhängen. Die Logdatei wird als Named Reference unter einem Dataset erzeugt.

7.2. Aufruf

```
tcpb_attach_file_to_job [-u=<> -p=<> [-g=<>]] -ds_type= -ds_name= [-nr_type=] -log= -job_uid= [-help]
```

7.3. Argumente

-u	= Login User
-p	= Login Password
-g	= Login Group
-ds_type	= Datasettyp
-ds_name	= Datasetname
-nr_type	= Named Reference Typ, wenn nicht gesetzt, wird ds_type benutzt
-log	= Logdatei, die angehängt wird (kompletter Pfad)
-job_uid	= Job ID
-h	= Ausgabe der Hilfe

7.4. Umgebung

Zum Start des Programms wird eine TC-Command Shell benötigt.

7.5. Dateien

Es werden die Standard TC und PLM-Easy Logfiles angelegt.

8. MAINTENANCE - TCPB_BATCH_RENAME

8.1. Beschreibung

Dieses Programm dient zum Umbenennen von Items, die in einer Eingabedatei definiert sind. Es ist als Steuerprogramm, das die interne PLM-Easy Item Rename Funktion aufruft, angelegt. Insofern gelten die für Item Rename in der TC Preference **TCPB_ItemRename** getroffenen Einstellungen analog (siehe TCPB_Portal+Core_V101.pdf)

8.2. Aufruf

```
tcpb_batch_rename [-u=<> -p=<> [-g=<>]] [-preserve_modification] [-bypass] -file=<input filename>
                  [-save_item] [-save_revs] [-separator=] [-comment=] [-h]
```

8.3. Argumente

-u	= Login User
-p	= Login Password
-g	= Login Group
-preserve_modification	= Keine Änderung der last_modified Einträge (user & date)
-bypass	= Programbearbeitung im bypass-mode, dba group erforderlich!
-file	= Eingabedatei (kompletter Pfad)
-save_item	= Item wird explizit nach der Umbenennung gespeichert
-save_revs	= Revisionen werden explizit nach der Umbenennung gespeichert
-separator	= Definiert ein alternatives Trennzeichen (Standardwert=";")
-comment	= Definiert ein alternatives Kommentarzeichen (Standardwert="#")
-h	= Ausgabe der Hilfe

Logfile name : <filename>.TCPB_BATCH_RENAME_ITEM.LOG

8.4. Umgebung

Zum Start des Programms wird eine TC-Command Shell benötigt. Siehe auch Teamcenter Help ⇒ Configuring Teamcenter ⇒ Preferences and Environment Variables Reference.

Da das Programm zur Laufzeit auf

```
libtcpb_basetool.dll
libtcpb_itktool.dll
libtcpb_se.dll
```

zugreift, muss in der PATH Variable ein Eintrag vorhanden sein, über den die DLLs gefunden werden können oder die DLLs müssen im Programmverzeichnis liegen.

8.5. Dateien

Eingabedatei: **test_tcpb_rename_item.dat**

```
# Example for TCPB_BATCH_RENAME_ITEM
#-----
#use following formats:
#
#<old item id>          <new item id>
# or
#<old item id>;<new item id>;
#
#<old item id>;<old rev id>;<new item id>;<new rev id>
# or
#<old item id>;<old rev id>;<new item id>;<new rev id>;
#
#<old item id>          ;<new item id>          ;[<new name>];[<new desc>]
#<old item id>;<old rev id>;<new item id>;<new rev id>;[<new name>];[<new desc>]
#
# different line formats (1,3 or 5 semicolons) can be mixed
#-----
Test000010      ;      Test000011
Test000011;;Test000012;;
Test000012;-003;Test000013;-004;TestPart;Die Beschreibung - darf 240 Zeichen haben
```

ADMIN TOOLS

```

Administrator: TC Command Prompt (64bit)
D:\Siemens\tc101_x64\portal>tcpb_batch_rename -h

Usage: tcpb_batch_rename [-u=<> -p=<> [-g=<>]] [-preserve_modification] [-bypass] -file=<input filename> [-h]

-u          = Login User
-p          = Login Password
-g          = Login Group
-preserve_modification = Avoid change of last_modified (user & date)
-bypass     = Run program in bypass-mode, dba group necessary!
-file       = Input filename
-h          = Prints out this message

Input file - line formats :

with one   semicolon: <old item id>;          <new item id>
with three semicolons: <old item id>;<old rev id>;<new item id>;<new rev id>
with five semicolons: <old item id>;          <new item id>;          ;[<new name>];[<new desc>]
with five semicolons: <old item id>;<old rev id>;<new item id>;<new rev id>;[<new name>];[<new des>]

where      <old item id>;<new item id>;;
has the same impact as <old item id>;<new item id>
and         <old item id>;<old rev id>;<new item id>;<new rev id>;;
has the same impact as <old item id>;<old rev id>;<new item id>;<new rev id>

Logfile name : <filename>.TCPB_BATCH_RENAME_ITEM.LOG

TCPB_Batch_Rename_Item V10.1.1.0

```

8.6. Zu Beachten

Die erlaubten Zeichenkettenlängen für Item-ID und Item-Name (128 Zeichen) sowie Description (240) und RevisionID (32) entsprechen denen der internen PLM-Easy Item Rename Funktion.

Die Parameter -u, -p und -g müssen immer zusammen oder überhaupt nicht (autologin) angegeben werden.
Der Parameter -bypass erfordert, dass der Anwender Bypass-Rechte besitzen muss (Mitglied der DBA-Gruppe).

Für bestimmte Aufgabenstellungen (z.B. Umbenennen von Revisionen und Datasets mit Status) sind entsprechende Berechtigungen nötig.
Dies kann erreicht werden durch:

- Freischalten der Zugriffsberechtigungen im Rahmen eines Freigabe-Prozesses.
Mit dem Handler EPM-set-rule-based-protection können die Zugriffsrechte auf die Target-Objekte dynamisch gesetzt werden.
- Nutzung von tcpb_batch_rename.exe mit einem ‚privilegierten‘ Benutzer.
Dazu muss für die User/Gruppen-Kombination die Berechtigung im AM-Ruletree angepasst werden.

8.7. Beispiele

```
tcpb_batch_rename -u=infodba -p=infodba -g=dba -file=c:\temp\test_tcpb_batch_rename_item.dat
```

mit autologin und ohne Änderung des last modified user und -date Eintrages:

```
tcpb_batch_rename -preserve_modification -file=c:\temp\test_tcpb_batch_rename_item.dat
```

mit alternativen Trenn- und Kommentarzeichen (z.B. falls diese in den Zeichenketten selbst vorkommen):

```
tcpb_batch_rename -separator=: -comment=! -file=c:\temp\test_tcpb_batch_rename_item.dat
```

mit explizitem Item Save (nur so können die am Item registrierten Post-actions ausgeführt werden):

```
tcpb_batch_rename -save_item -file=c:\temp\test_tcpb_batch_rename_item.dat
```

9. MAINTENANCE - TCPB_CHANGE_OWNERSHIP

9.1. Beschreibung

Dieses Programm ermöglicht den Besitzerwechsel für Items und die zugehörigen Workspace Objekte. Optionen erlauben einen gezielten oder umfassenden Besitzerwechsel.

9.2. Aufruf

```
tcpb_change_ownership -u=<user> -p=<passwd> -g=<group>  
                      -file=<input_datei> -folder=<input_folder> -log=<logfile>  
                      -to_user=<new_user> -to_group=<new_group>  
                      -from_user=<old_user> -from_group=<old_group>  
                      -h
```

9.3. Argumente

-u	= Login User
-p	= Login Passwort
-g	= Login Gruppe
-log	= Protokoll mit den geänderten Items
-file	= Datei (kompletter Pfad) enthält ItemIds (bzw. ItemId;RevisionId Paare), welche neuen Besitzer (Gruppe) erhalten sollen. Nur diese Items werden bearbeitet.
-folder	= TC Folder enthält Items, die neuen Besitzer und/oder neue Gruppe erhalten sollen. Nur diese Items werden bearbeitet. Der Login User muß Besitzer dieses Verzeichnisses sein!
-from_user	= Aktueller Besitzer (nicht genutzt bei -folder bzw. -file)
-from_group	= Aktuelle Gruppe (nicht genutzt bei -folder bzw. -file)
-to_user	= Neuer Besitzer
-to_group	= Neue Gruppe
-h -help	= Ausgabe der Hilfe


```

Administrator: TC Command Prompt (64bit)

D:\Siemens\tc101_x64\portal>D:\SiemensShare\prodlocal\bin64\tcpb_change_ownership.exe -h
tcpb_change_ownership V10.1.1.0

Usage: D:\SiemensShare\prodlocal\bin64\tcpb_change_ownership.exe
-u=<user>
-p=<password>
-g=<group> -- You must have bypass privileges.
-log=<logfile>
      :if -logfile not set, tcpb_change_ownership.log is written to the current temp directory
-to_user=<new_owner>
      :if -to_user not set, owning user of the object will be retained
-to_group=<new_group>
      :if -to_group not set, owning group of the object will be retained

case 1: search only for items
-from_user=<old_owner>
-from_group=<old_group>
      :Note! Both options are mandatory!

case 2:
-folder=<input_folder>
      :if input_folder, then only the Items in the input_folder will be attached!
      :Note! Login user must be owner of input_folder!

case 3:
-file=<input file>
      :if input_file, then only the Items in the input_file will be attached!

      :# Input file - line formats:
      :# one ItemId per line - owner of item and subsequent objects will be changed
      :# one ItemId;RevId pair per line - owner of item revision will be changed
      :# in case of RevId - use of ; is mandatory
      :<item_id_1>
      :<item_id_2>
      :<item_id_3>;<rev_id_1>
      :<item_id_3>;<rev_id_2>
      :....

      -hfe!l!p! : this output.

D:\Siemens\tc101_x64\portal>

```

9.4. Umgebung

Zum Start des Programms wird eine Tc-Command Shell benötigt.

Siehe auch Teamcenter Help ⇒ Configuring Teamcenter ⇒ Preferences and Environment Variables Reference.

9.5. Dateien

Es werden die Standard TC Log files angelegt, siehe auch Teamcenter Help ⇒ Administering Teamcenter ⇒ Utilities Reference ⇒ Getting started ⇒ Log files.

ADMIN TOOLS

9.6. Beispiel

```
tcpb_change_ownership -u=infodba -p=<pw> -g=dba -to_user=ugsplm1 -to_group=dba file=<path>\owner_test.dat
```

The screenshot shows the Teamcenter interface. On the left, a file named 'owner_test.dat' is open, containing two lines of text: '1 DEMO000009;- ' and '2 S000078'. On the right, the 'S000078-001' object is selected, and its details are displayed in a table.

object_string	object_type	Relation	owning_user	ownin...	last_mod_date	check
S000078	STD Master	IMAN_master_form	ugsplm1 (ugsplm1)	dba	18-Jan-2013 15:31	
S000078/-;1-001	STD Revision	revision_list	ugsplm1 (ugsplm1)	dba	18-Jan-2013 15:31	

The screenshot shows the 'tcpb_change_ownership.log' file in Notepad. The log contains the following text:

```
Protocol of tcpb_change_ownership V8.3.3.3
The Owner and Group of the following Workspace Objects were changed:
```

nr	object	type	owner	group
00001	DEMO000009/-	TCPBDemo Revision	ugsplm1	dba
00002	S000078/-	STD Revision	ugsplm1	dba
00003	S000078	STD	ugsplm1	dba

9.7. Zu Beachten

- Für den Case 1 (Suche nach Items) müssen sowohl `-from_user` als auch `-from_group` gesetzt sein.
- Falls entweder `-to_user` oder `-to_group` nicht gesetzt sind, wird der Besitzer bzw. die Gruppe nicht geändert.
- Für die Berücksichtigung der Dokumente müssen die folgenden Relationen in den TC - Preferences gesetzt sein:

```
# Change Owner
TCPB_change_owner_group_relation=
IMAN_specification
IMAN_requirement
IMAN_UG_altrep
```

10. MAINTENANCE - TCPB_CHANGE_VOLUME

10.1. Beschreibung

Dieses Programm dient zum automatischen Modifizieren von Volumedefinitionen, zum Verschieben von Volumes und zum Modifizieren des Zugriffs für User und Gruppen.

10.2. Aufruf

```
tcpb_change_volumes [-u=<user id>] [-p=<passwd>] [-g=<group>] -mode=[list|modify] -file=<filename> [-access]
[-log=<logfile>] [-dryrun] [-help]
```

10.3. Argumente

-u	= Login User
-p	= Login Passwort
-g	= Login Gruppe
-mode	= list: zur Auflistung aller Volumedefinitionen nach -file modify: zur Modifikation von Volumes entsprechend -file
-file	= je nach Modus die Ein- oder Ausgabedatei
-log	= Logdatei
-access	= je nach Modus wird der Zugriff für User und Gruppen aufgelistet oder modifiziert
-dryrun	= Testlauf - es wird nichts in der Datenbank geändert
-h -help	= Ausgabe der Hilfe

10.4. Umgebung

Zum Start des Programms wird eine TC Command Shell benötigt.

Siehe auch Teamcenter Help ⇒ Configuring Teamcenter ⇒ Preferences and Environment Variables Reference.

ADMIN TOOLS

```

Administrator: TC Command Prompt (64bit)
*****
** Copyright (c) 2000-2012 Siemens PLM Software
** tcpb_change_volumes V10.1.1.0
*****

Usage: tcpb_change_volumes [-u=] [-p=] [-g=] [-mode=list|modify] [-file=] [-log=] [-access] [-help] [-dryrun]
tcpb_change_volumes V10.1.1.0

-u          = Login User (needs -u/-p/-g or none)
-p          = Login Password (needs -u/-p/-g or none)
-g          = Login Group (needs -u/-p/-g or none)
-mode       = list: writing infos about available volumes to -file
              modify: modifying volumes according to -file
              Default is -mode=list
-file       = Output-file when using -mode=list, input-file when using -mode=modify
-log        = Log-File
-dryrun     = No action will be performed
-access     = List (modify) access to volumes.
              In -mode=list users granted access to volumes and groups granted or revoked are listed.
-help       = Prints out this message

Input file format -access NOT given:
<volume_name>;<node_name>;<node_type: Unix|Windows>;<unix_path>;<winnt_path>;<move_volume: Y|y|N|n>
  -> Raw input file could be generated by using this program with -mode=list
  -> move_volume could be N|n for modify volume-data or Y|y for move volume
  -> First character # in input-line marks comment-lines

Input file format -access given:
<volume_name>;[GROUP|USER];[<groupname>|<userId>];[grant|revoke]
  -> Example: volume1;USER;infodba;grant
  -> First character # in input-line marks comment-lines

Notes:
The group of the executing user has to be dba!
For modifying volumes (move_volume N|n) the (new) volume directory must exist!
For moving volumes (move_volume Y|y) the new volume directory should not exist!
The volume-name could not be modified!

Log Out all users before starting in -mode=modify!

```

10.5. Dateien

10.5.1. Eingabedatei

Es muss unterschieden werden, ob der Schalter `-access` gesetzt ist oder nicht.

ADMIN TOOLS

KEIN –access gesetzt:

Das Inputfile enthält Zeilen mit folgender Syntax:

<volume name>;<node name>;<node type>;<unix path>;<winnt path>;<move volume>

Feld	Bedeutung
<volume_name>	Name des Volumes in TC
<node_name>	Knotenname, d.h. Rechnername des imanfs Servers welche für die Verwaltung des Volumes zuständig ist
<node_type>	Knotentyp des imanfs Rechners. Dieses Feld kann die Werte „Windows“ für Windows Rechner oder „Unix“ für Unix Rechner annehmen.
<unix_path>	Pfadangabe für Unix Volumes – neuer Zielpfad
<winnt_path>	Pfadangabe für Windows Volumes – neuer Zielpfad
<move_volume>	Diese Feld kann die Werte „Y“, „y“ für Yes und „N“, „n“ für No annehmen. Yes bedeutet, dass das Volume incl. aller Files und Verzeichnisse verschoben wird. No bedeutet, dass nur die Volumeinformationen editiert werden.

Zeilen mit “#” als erstem Zeichen werden als Kommentare interpretiert.

Beispiel

Volume01;Server;Windows;;D:\ugsshare\tc_volumes\volume01;N

Volume02;Server;Windows;;D:\ugsshare\tc_volumes\volume02;N

Volume03;Server;Windows;;D:\ugsshare\tc_volumes\volume03;N

Schalter –access gesetzt:

Feld	Bedeutung
<volume_name>	Name des Volumes in TC
[GROUP USER]	Zeiger dafür, ob im nächsten Feld der Name eines Users oder einer Gruppe angegeben ist
[<groupname> <userid>]	Name der Gruppe oder des Users
[grant revoke]	Soll ein Grant oder Revoke durchgeführt werden?

Zeilen mit “#” als erstem Zeichen werden als Kommentare interpretiert.

Beispiel

voltest;USER;peter;grant

voltest;GROUP;Projekt1;grant

voltest;GROUP;system;grant

voltest;GROUP;dba;revoke

voltest2;USER;lisauser;grant

voltest2;USER;infodba;grant

voltest2;GROUP;dba;grant

voltest2;GROUP;system;grant

voltest2;GROUP;Projekt1;revoke

10.5.2. Ausgabedatei

Die Ausgabedatei ist in der gleichen Syntax wie die Eingabedatei, wenn der Schalter –access nicht gesetzt ist, fehlt hier nur die Spalte <move_volume>. Ist der Schalter –access gesetzt, werden zwar alle Gruppen aufgelistet (revoke oder grant), bei den Usern werden aber nur die aufgeführt, die einen „grant“ Zugriff haben.

10.5.3. Logdatei

In der Logdatei werden alle durchgeführten Änderungen protokolliert. Wird die Option „-logfile“ nicht angegeben, so werden diese Informationen nach Standard Output protokolliert.

10.5.4. Sonstige Files

Es werden die Standard TC Log files angelegt, siehe auch Teamcenter Help ⇨ Administering Teamcenter ⇨ Utilities Reference ⇨ Getting started ⇨ Log files.

10.6. Zu Beachten

Alle Änderungen sollten zwingend noch einmal in TC nachgeprüft werden!

Änderungen sollten nur dann durchgeführt werden, wenn keine Anwender in der Datenbank arbeiten. (sperren der DB mit install –lock_db –u= –p= –g=)

Dieses Utility muss unter einem Benutzer der Gruppe „dba“ gestartet werden.

Die Parameter -u, -p und -g müssen immer zusammen oder überhaupt nicht (autologin) angegeben werden.

Werden nur die Volumeinformationen geändert, ohne dass Volume zu verschieben (Spalte: <move_volume> = N), dann muss das Zielverzeichnis schon existieren.

Wird hingegen das Volume mit Inhalt verschoben (Spalte: <move_volume> = Y), dann darf das Zielverzeichnis nicht existieren.

Der Volumenname kann nicht geändert werden.

10.7. Beispiele

Das Protokoll zum Checkin inklusive einer Zusammenfassung wird in die Standard-Ausgabe geschrieben.

10.7.1. Beispiel 1: Listen aller Volumeinformationen

Kommando:

```
tcpb_change_volumes -u=infodba -p=***** -g=dba -mode=list -file=volumeinfo.txt
```

Ausgabedatei:

Volume01;Server;Windows;;D:\ugsshare\tc_volumes\volume01

Volume02;Server;Windows;;D:\ugsshare\tc_volumes\volume02

Volume03;Server;Windows;;D:\ugsshare\tc_volumes\volume03

ADMIN TOOLS

10.7.2. Beispiel 2: Ändern aller Volumeinformationen von „tc_volumes“ nach „test_volumes“Manuelle Arbeit 1:

Erst einmal müssen die entsprechende Volumeverzeichnisse wie folgt kopiert werden:

D:\ugsshare\tc_volumes\volume01 nach D:\ugsshare\test_volumes\volume01

D:\ugsshare\tc_volumes\volume01 nach D:\ugsshare\test_volumes\volume01

D:\ugsshare\tc_volumes\volume01 nach D:\ugsshare\test_volumes\volume01

Manuelle Arbeit 2:

Die Inputdatei kann nun wie folgt aus dem Beispiel (2.7.1.) übernommen und editiert werden:

Volume01;Server;Windows;;D:\ugsshare\test_volumes\volume01;N

Volume02;Server;Windows;;D:\ugsshare\test_volumes\volume02;N

Volume03;Server;Windows;;D:\ugsshare\test_volumes\volume03;N

Kommando:

```
tcpb_change_volumes -u=infodba -p=***** -g=dba -mode=modify -file=volumeinfo.txt
```

Wichtig:

Nach dieser Aktion unbedingt einloggen in TC und prüfen, ob die Volumes nun richtig erkannt werden.

10.7.3. Beispiel 3: Verschieben eines Volumes von „D:“ nach „E:“Manuelle Arbeit 1:

Die Inputdatei wird wie folgt aus dem Beispiel (2.7.1.) übernommen und editiert:

#Volume01;Server;Windows;;D:\ugsshare\tc_volumes\volume01;N

#Volume02;Server;Windows;;D:\ugsshare\tc_volumes\volume02;N

Volume03;Server;Windows;;E:\ugsshare\tc_volumes\volume03;Y

Im o. g. Beispiel sind die ersten beiden Zeilen auskommentiert (#-Zeichen) und Volume03 wird von D:\ugsshare\tc_volumes\volume03 nach E:\ugsshare\tc_volumes\volume03 verschoben.

Kommando:

```
tcpb_change_volumes -u=infodba -p=***** -g=dba -mode=modify -file=volumeinfo.txt
```

Wichtig:

Nach dieser Aktion unbedingt einloggen in TC und prüfen, ob die Volumes nun richtig erkannt werden.

10.7.4. Beispiel 4: Listen der ZugriffseigenschaftenKommando:

```
tcpb_change_volumes.exe -u=infodba -p=***** -g=dba -mode=list -file=out3.txt -access -log=log2.txt
```

Wichtig:

Nach dieser Aktion unbedingt einloggen in TC und prüfen, ob der Access nun richtig erkannt wird.

10.7.5. Beispiel 5: Ändern der Zugriffseigenschaften

Auch hier kann man die vorher nach Beispiel 4 erstellte Datei editieren. Je nach Vorgehensweise bei der Erstellung der User/Gruppen ist den User/Gruppen ein Standardvolumen zugewiesen, worunter sich dann ein entsprechendes Verzeichnis befindet. Um eine neues Verzeichnis für den User oder die Gruppe in einem der Volumes zu erzeugen, ist es zum Beispiel möglich, zunächst ein „Revoke“ durchzuführen und dann ein „Grant“.

Kommando:

```
tcpb_change_volumes.exe -u=infodba -p=***** -g=dba -mode=modify -file=out3.txt -access -log=log2.txt
```

Datei:

Im ersten Schritt wird die Datei ,wie in Beispiel 4 beschrieben, ausgegeben, um diese Datei dann in einem nächsten Schritt zu editieren:

```
voltest;USER;peter;revoke  
voltest;GROUP;Projekt1;revoke  
voltest;GROUP;system;revoke
```

Jetzt mit obigem Kommando ein Revoke durchführen, dann die Datei erneut editieren:

```
voltest;USER;peter;grant  
voltest;GROUP;Projekt1;grant  
voltest;GROUP;system;grant
```

Und nun mit demselben Kommando (besser den Namen der Logdatei ändern) das Grant vollziehen.

Wichtig:

Nach dieser Aktion unbedingt einloggen in Teamcenter und prüfen, ob der Access nun richtig erkannt wird.

11. MAINTENANCE - TCPB_CHECKIN_BY_CONFIG

11.1. Beschreibung

Dieses Programm wurde mit TCPB81 neu eingeführt und dient zum automatischen Einchecken von implizit oder explizit ausgecheckten Datensätzen und BOMview Revisions (Stücklisten) einzelner Benutzer oder Benutzergruppen.

Es ersetzt TCPB_AUTO_CHECKIN, welches ab NX6.0.4 wegen geänderter Change Reasons nicht mehr korrekt funktioniert. Eine Zielstellung bei der Programmentwicklung war die Konfigurierbarkeit, um von zukünftigen Änderungen der Change Reasons unabhängig zu sein.

Die Change Reasons werden aus einer Konfigurationsdatei eingelesen und können somit jederzeit angepasst werden.

11.2. Aufruf

```
tcpb_checkin_by_config [-u=] [-p=] [-g=] <-user=|-group=> [-all] [-dryrun] [-cfg=] [-help]
```

11.3. Argumente

-u	= Login User
-p	= Login Passwort
-g	= Login Gruppe
-user	= User, der Objekte ausgecheckt hat
-group	= Gruppe, die Objekte ausgecheckt hat
-all	= Checkt alle Objekte ein (cfg Einstellungen werden ignoriert)
-dryrun	= Testlauf - es werden keine Objekte eingecheckt
-cfg	= Konfigurationsdatei (kompl. Pfad) mit checkout reasons (default, falls nicht gesetzt: <tcpb_config_path>/tcpb_CheckinCfg.ini)
-h -help	= Ausgabe der Hilfe

Ein Einchecken aller Checkouts wird mit der Option -all durchgeführt. In diesem Fall wird die Konfigurationsdatei ignoriert.

11.4. Umgebung

Zum Start des Programms wird eine TC-Command Shell benötigt.

Siehe auch Teamcenter Help ⇨ Configuring Teamcenter ⇨ Preferences and Environment Variables Reference.

Falls TCPB_**** Umgebungsvariablen benötigt werden, wird die Benutzung einer TCPB-Command Shell empfohlen.

11.5. Dateien

Die Konfigurationsdatei wird entweder mit dem Parameter `-cfg=<kompletter Pfad>` definiert oder es wird eine Datei `tcpb_CheckinCfg.ini` in einem Standardkonfigurationsverzeichnis erwartet (`%TCPB_CONFIG%` oder `%TCPB_LOCAL%\config`). Falls der Parameter `-all` gesetzt ist, wird keine Konfigurationsdatei ausgewertet.

Eine Beispieldatei `tcpb_CheckinCfg.ini` liegt in der PLM-Easy Distribution unter `%TCPB_LOCAL%\config`

Es werden die Standard TC und PLM-Easy Logfiles angelegt.

11.6. Zu Beachten

Achtung:

Checken Sie keine Daten ein, auf denen ein Anwender gerade aktiv arbeitet !

Das Programm muss unter einem Benutzer der Gruppe „dba“ gestartet werden.

Die Parameter `-u`, `-p` und `-g` müssen immer zusammen oder überhaupt nicht (autologin) angegeben werden.

Die Parameter `-user` und `-group` beziehen sich nicht auf das Workspace-Objekt (was ausgecheckt wurde), sondern auf Reservation Objekt (= User und aktuelle Gruppe von dem, der ausgecheckt hat).

11.7. Konfigurationsdatei

In der Konfigurationsdatei können die für das Checkin zu beachtenden Objekte genauer spezifiziert werden. Für jeden Change Reason ist unter [CHECKINOPTION] eine Zeile anzugeben.

Die Anzahl der Zeilen muß mit dem Wert für REASONCOUNT übereinstimmen.

Die ChangeID wird zurzeit nicht ausgewertet, da der Change Reason ab NX6.0.4 eindeutig ist. Der Parameter ChangeID ist für eine zukünftige Nutzung reserviert.

Soll für alle Change Reasons eingcheckedt werden, kann dies mit wildcard (*) erfolgen (siehe Beispielkonfiguration).

In diesem Fall wird zusätzlich ein NX-Flag ausgewertet, der die Auswahl der Datasets steuert:

NX-flag

0 = Dataset Typ wird nicht getestet, d.h. alle Datasets (default)

1 = nur NX Datasets

2 = Alle Datasets außer NX

```
#=====
#
#      Copyright (c) 2003-2012 Siemens Industry Software
#      Unpublished - All Rights Reserved
#
#=====
#
# File name   : tcpb_CheckinCfg.ini
# Description : Configuration file of tcpb_checkin_by_config command.
#              (TCPB standalone application to check-in objects
#              automatically. This is the replacement for tcpb_auto_checkin
#              starting with NX6.0.4)
```

ADMIN TOOLS

```
#
#
# Default path: %TCPB_CONFIG%\tcpb_CheckinCfg.ini
# or use "-cfg=" option of tcpb_checkin_by_config command
#=====
#
# REASONCOUNT=<value>          ...Define number of entries (lines) below [CheckinOption]
#
[General]
REASONCOUNT=1
#
#
# The lines below [CheckinOption] will define which objects have to be checked-in.
# There has to be a separate line for each Change Reason.
# The format is "Change ID;Change Reason;NX-flag"
# Semicolon separators are mandatory.
#
# Change ID is not used/tested (for future use). The entry is mandatory. It could be set to "*" or " ".
#
# The Change Reason is unique for the different implicate checkout types.
# If all Change Reasons should be used for check-in, set "*". This will include explicite checkouts too.
#
# Examples for Change Reason:
#   NX Structure Checkout
#   Locking BVRs
#   Locking remote BVRs
#   ANT from Unigraphics
#   Locking Parents for Save-As in NX
#   Lock-On-Modify from NX
#   Modified Parts from NX
#   NX Structure Checkout
#   Try_To_Lock_Part from NX
#   UGMGR_i_ant_check_out from NX
#   NX Save
#   Lock reprev from NX
#   CheckOutSecondaryDataset from Unigraphics
#
# The NX-flag is used to differentiate between NX and non-NX datasets.
# It will be tested in case of Change Reason is set to "*" only.
# NX-flag can be set to:
#   0 - do not check dataset type (default),
#   1 - NX datasets only,
#   2 - all datasets, except NX
#
[CheckinOption]
#Change ID;Change Reason;NX-flag
*;*;1
```

11.8. Beispiele

Es gibt zwei Haupteinsatzfälle für dieses Programm. Das erste Szenario beinhaltet das Wieder-Einchecken von Hinterlassenschaften abgestürzter oder getrennter NX Sessions. Für diese sogenannten impliziten Checkouts ist die Angabe des Change Reasons notwendig. Diese internen Change Reasons sind seit NX6 Veränderungen unterworfen. Die bekannten Change Reasons sind in der Beispielkonfigurationsdatei tcpb_CheckinCfg.ini als Kommentar enthalten.

Einchecken aller impliziten Checkouts der Gruppe **training**:

```
tcpb_checkin_by_config -u=admin -p=*** -g=dba -group=training -cfg=x:\admin\config\implizit_auto_checkin.cfg
```

Konfigurationsdatei: [x:\admin\config\implizit_auto_checkin.cfg](#)

```
[General]
REASONCOUNT=13

[CheckinOption]
*;NX Structure Checkout;0
*;Locking BVRs;0
*;Locking remote BVRs;0
*;ANT from Unigraphics;0
*;Locking Parents for Save-As in NX;0
*;Lock-On-Modify from NX;0
*;Modified Parts from NX;0
*;NX Structure Checkout;0
*;Try_To_Lock_Part from NX;0
*;UGMGR_i_ant_check_out from NX;0
*;NX Save;0
*;Lock reprev from NX;0
*;CheckOutSecondaryDataset from Unigraphics;0
```

Achtung:

Der Change Reason **ANT from Unigraphics** wird durch ein Checkout im NX Baugruppennavigator (ANT) generiert, und ist von NX aus gesehen, ein expliziter Checkout, der auch nach dem korrekten Schließen von NX in TC erhalten bleibt. Es ist ev. nicht gewünscht, diese Datensätze mit einzuchecken. Dann sollte in der Konfigurationsdatei die Zeile mit **ANT from Unigraphics** entfallen.

Sollten Probleme mit den Change Reasons auftreten, empfiehlt es sich, das Kommando mit `--dryrun` und `--all` und mit eingeschaltetem TRACE (siehe Basic Installation) laufen zu lassen. Die Logdatei enthält dann alle gefundenen Change Reasons. Ebenso können diese Informationen aus den Meldungen in der CMD-Shell entnommen werden.

```
*****
** checkin_by_config Version 8.3.2.2
   running tce-version:Teamcenter V8000.3.0.42_20110823.00
Set bypass for dba user!
Number of checked-out elements      : 2
=====

----- <0> -----
----> Was not loaded
```

ADMIN TOOLS

```
Element ID      : S000004_-003
Element Name    : S000004_-003
Element Beschreibung:
Element Typ     : UGMASTER
Change Id       :
Change Reason   : Lock-On-Modify from NX
-----
Element ID      : S000004_-003
Element Name    : S000004_-003
Element Beschreibung:
Element Typ     : UGMASTER
Element Besitzer : infodba
Element Gruppe  : dba
Element Erstelldatum: 02-Sep-2011 16:26
-----
Change Id       :
Change Reason   : Lock-On-Modify from NX
-> Element would be checked-in (DRYRUN).
```

Das zweite Szenario wäre die Vorbereitung eines Softwareupgrades bzw. einer Datenmigration. Dann soll alles eingchecked werden.

```
tcpb_checkin_by_config -u=admin -p=*** -g=dba -group=users -all
```

oder

```
tcpb_checkin_by_config -u=admin -p=*** -g=dba -group=users -cfg=x:\admin\config\auto_checkin.cfg
```

Konfigurationsdatei: [x:\admin\config\auto_checkin.cfg](#)

```
[General]
REASONCOUNT=1

[CheckinOption]
*;*;0
```

12. MAINTENANCE - TCPB_INIT_SEED_COUNTER

12.1. Beschreibung

Dieses Programm ermöglicht es, den Startwert für die ItemID zu setzen.

12.2. Aufruf

```
tcpb_init_seed_counter [-u=<user> -p=<password> -g=<group>] -seed_counter=<seed counter name>  
[-init_val=<initial value> | -show_val] [-h]
```

12.3. Argumente

-u	= Login User
-p	= Login Passwort
-g	= Login Gruppe
-seed_counter	= Name des Nummerngenerators
-init_val	= Startwert für den Nummerngenerator
-show_val	= zeigt den aktuellen Wert des Nummerngenerators an
-h	= Ausgabe der Hilfe

12.4. Umgebung

Zum Start des Programms wird eine TC-Command Shell benötigt. Siehe auch Teamcenter Help ⇨ Configuring Teamcenter ⇨ Preferences and Environment Variables Reference.

12.5. Beispiel

Für das Setzen des Startwertes des Nummerngenerators für den Itemtype „STD“ – siehe tcpb_base.ini

Kommando:

```
tcpb_init_seed_counter -u=infodba -p=infodba -g=dba -seed_counter=STD_item_id -init_val=2000
```

13. MAINTENANCE - TCPB_PURGE_DATASET

13.1. Beschreibung

Dieses Programm dient zum Bereinigen von Dataset Versionen (nur die neueste bleibt erhalten). Dies kann z.B. nach einem NX Refile notwendig sein. Die zu bereinigenden Datasets werden mithilfe einer Eingabedatei definiert.

13.2. Aufruf

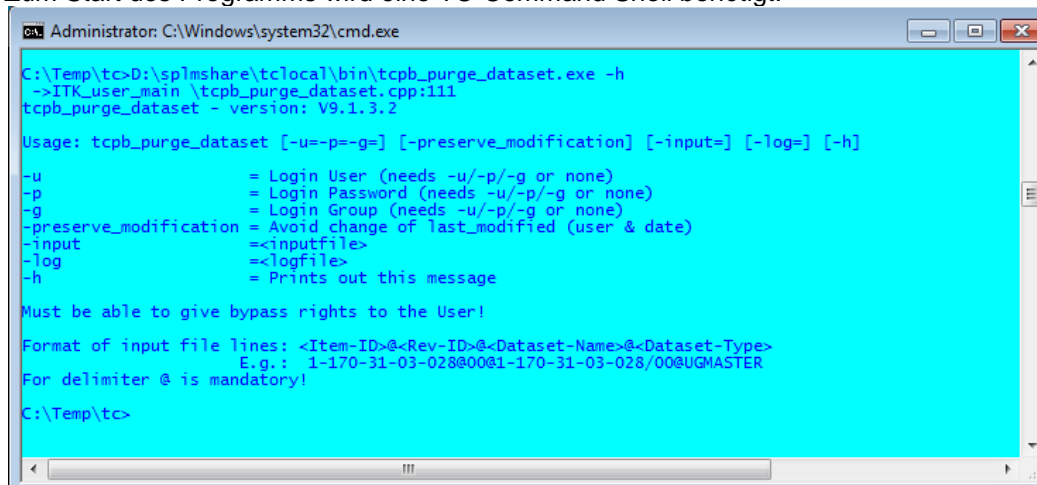
```
tcpb_purge_dataset [-u=<> -p=<> [-g=<>]] [-preserve_modification] -input=<input filename> [-log=] [-help]
```

13.3. Argumente

-u	= Login User
-p	= Login Password
-g	= Login Group
-preserve_modification	= Keine Änderung der last_modified Einträge (user & date)
-input	= Eingabedatei (kompletter Pfad)
-log	= log file, default: c:\temp\TCPB_purge_dataset.log
-h help	= Ausgabe der Hilfe

13.4. Umgebung

Zum Start des Programms wird eine TC-Command Shell benötigt.



```
Administrator: C:\Windows\system32\cmd.exe

C:\Temp\tc>D:\splmshare\tclocal\bin\tcpb_purge_dataset.exe -h
->ITK_user_main\tcpb_purge_dataset.cpp:111
tcpb_purge_dataset - version: V9.1.3.2

Usage: tcpb_purge_dataset [-u=-p=-g=] [-preserve_modification] [-input=] [-log=] [-h]

-u           = Login User (needs -u/-p/-g or none)
-p           = Login Password (needs -u/-p/-g or none)
-g           = Login Group (needs -u/-p/-g or none)
-preserve_modification = Avoid change of last_modified (user & date)
-input       = <inputfile>
-log         = <logfile>
-h           = Prints out this message

Must be able to give bypass rights to the User!

Format of input file lines: <Item-ID>@<Rev-ID>@<Dataset-Name>@<Dataset-Type>
E.g.: 1-170-31-03-028@0081-170-31-03-028/00@UGMASTER
For delimiter @ is mandatory!

C:\Temp\tc>
```

13.5. Dateien

Eingabedatei: **test_tcpb_purge_dataset.dat**

```
# Example for TCPB_purge_dataset
#-----
# Format of input file lines:
#<Item-ID>@<Rev-ID>@<Dataset-Name>@<Dataset-Type>
#
#
#-----
E00000035@000@E00000035_000@UGMASTER
```

Logdatei: **test.log**

```
Logfile: tcpb_purge_dataset - version: V9.1.3.2
INPUT: D:\splmshare\tclocal\bin\test_tcpb_purge_dataset.dat
LOGFILE: D:\splmshare\tclocal\bin\test.log
-----

ItemId [E00000035], RevId [000], Name[E00000035_000], Type[UGMASTER]
1.) ItemId [E00000035] RevId [000] Name [E00000035_000] purged.

=====
End of tcpb_purge_dataset
=====
```

13.6. Zu Beachten

Die Ausführung dieses Programms erfordert DBA Rechte!

Die erlaubte Länge für die Zeilen der Eingabedatei beträgt 200 Zeichen.

Die Parameter -u, -p und -g müssen immer zusammen oder überhaupt nicht (autologin) angegeben werden.

13.7. Beispiel

Kommando:

```
tcpb_purge_dataset -u=infodba -p=infodba -g=dba -file=c:\temp\test_tcpb_purge_dataset.dat
```

mit autologin:

```
tcpb_purge_dataset -preserve_modification -file=c:\temp\test test_tcpb_purge_dataset.dat
```


14. MAINTENANCE - TCPB_SET_ATTR_VALUE

14.1. Beschreibung

Dieses Programm dient dazu, ein Attribut einer Item Master oder Item Revision Master Form zu setzen.

14.2. Aufruf

```
tcpb_set_attr_value [-u=<> -p=<> [-g=<>]] <-item=> [-rev=] -dest=<IRM|IM> -attr_name= [-attr_value=]  
[-no_overwrite] [-help]
```

14.3. Argumente

-u	= Login User
-p	= Login Password
-g	= Login Group
-item	= Item ID
-rev	= Revision ID, falls nicht angegeben, wird automatisch die letzte ItemRevision verwendet
-dest	= <IRM IM> IRM = ItemRevision Masterform IM = Item Masterform
-attr_name	= Attributname
-attr_value	= Attributwert
-no_overwrite	= falls ein alter Wert vorhanden ist, wird er nicht überschrieben
-h	= Ausgabe der Hilfe

14.4. Umgebung

Zum Start des Programms wird eine TC-Command Shell benötigt.

14.5. Dateien

Es werden die Standard TC und PLM-Easy Logfiles angelegt.

DATA-EXCHANGE FUNKTIONEN

15. EXCHANGE - TCPB_DATA_IMPORT

Dieses Programm dient zum Datenimport aus .csv Dateien. Wegen der umfangreichen Konfigurationsmöglichkeiten sind alle Informationen in einer eigenen Dokumentation zusammengefasst (siehe **TCPB_Data_Import_V101.pdf**).

16. EXCHANGE - TCPB_EXPORT_FILE

16.1. Beschreibung

Dieses Programm ermöglicht den Export von Dateien aus Datasets, d.h. der „Benannten Referenzen“. Dabei wird auch eine Informationsdatei mit Attributen zum Item, der ItemRevision, dem Dataset und der zu exportierenden Datei geschrieben.

16.2. Aufruf

```
tcpb_export_file -u= -g= -p= [-vb] [-replace] (-i=|-f=) [-item= [-rev=] -d= [-rel=] -type= -ref=
-export={y|n}] [-itemInf]
```

16.3. Argumente

```
-u          = <Login User>
-p          = <Login Password>
-g          = <Gruppe>
-i          = <Eingabedatei für Mehrfachexport>
-f          = <Exportdatei, Einzelexport>
-vb         = verbose mode, d.h. Ausgabe von Informationen neben eventuellen Fehlermeldungen
-replace    = Dateien ersetzen, falls schon im Filesystem vorhanden
-d          = <Datensatzname> - Name des zu exportierenden Datensatzes
-rel        = <Relation> - TC Relation des Datensatzes zur ItemRevision
-type       = <Datensatz Typ> - Typ des zu exportierenden Datensatzes
-ref        = <NamedReference Typ> - Typ der zu exportierenden „Benannten Referenz“
-export     = {y|n} y = Export der Datei und Attribute, n = nur Export der Attribute (siehe -itemInf)
-item       = <Item ID>
-rev        = <Revision ID>
-itemInf    = <Attributedatei> - Informationen zu Item, ItemRevision und Dataset
-h/-help    = Ausgabe der Hilfe
```

ADMIN TOOLS

Hinweis:

Es muss entweder die Option `-i` oder die Option `-f` gesetzt sein. Wird die Option `-i` gewählt, muß die `<Eingabedatei>` die folgenden Parameter enthalten. Für die Option `-f` sind diese Parameter in der Befehlszeile anzugeben.

```
-replace = Dateien ersetzen, falls schon im Filesystem vorhanden
-d       = <Datensatzname> - Name des zu exportierenden Datensatzes
-rel     = <Relation> - TC Relation des Datensatzes zur ItemRevision
-type    = <Datensatz Typ> - Typ des zu exportierenden Datensatzes
-ref     = <NamedReference Typ> - Typ der zu exportierenden „Benannten Referenz“
-export  = {y|n} y = Export der Datei und Attribute, n = nur Export der Attribute
          (siehe -itemInf)
-item    = <Item ID>
-rev     = <Revision ID>
-itemInf = <Attributedatei> - Informationen zu Item, ItemRevision und Dataset
```

16.4. Umgebung

Zum Start des Programms wird eine TC-Command Shell benötigt.

Siehe auch Teamcenter Help ⇒ Configuring Teamcenter ⇒ Preferences and Environment Variables Reference.

16.5. Dateien

Es werden die Standard TC Log files angelegt, siehe auch Teamcenter Help ⇒ Administering Teamcenter ⇒ Utilities Reference ⇒ Getting started ⇒ Log files.

16.6. Beispiel

Es wird ein Text Dokument (Datensatz: Test000046_-) aus der ItemRevision Test000046/-, exportiert.

```
tcpb_export_file -u=infodba -p=infodba -g=dba -item=Test000046 -rev=- -rel=IMAN_specification
-type=Text -d=Test000046_- -ref=Text -export=y -itemInf=c:\temp\test000046.inf -f=c:\temp\test000046.txt -vb
```

Ausgabe (durch Verbose Mode `-vb`):

```
Startup ok!
Bypass set. You are a member of group dba.
```

```
New Export starts here
Export File : c:\temp\test.prt
Item ID     : Test000046
Rev ID      : -
Dataset Name: Test000046_-_Z01
Dataset Relation: IMAN_specification
```

ADMIN TOOLS

```
Dataset Type: UGPART
NR Type      : UGPART
Item Inf     : c:\temp\info.txt
export       : yes
replace      : no
Status       : Export successfull.
```

Exiting program with error code : <0>

Attributedatei: c:\temp\test000046.inf

```
Item.name=Export Test
Item.type=Item
Item.id=Test000046
Item.desc=Test Item
Item.owner=ugsplm1
Item.group=dba
Item.created=05-Jan-2010 10:35
Item.modified=23-Mar-2010 11:04
Item.released=
Item.last_rel_stats=
Item.owning_site=
Revision.name=Export Test
Revision.type=ItemRevision
Revision.id=Test000046/-
Revision.desc=Test ItemRevision
Revision.owner=ugsplm1
Revision.group=dba
Revision.created=05-Jan-2010 10:35
Revision.modified=23-Mar-2010 11:04
Revision.released=
Revision.last_rel_stats=
Dataset.relation=Specification
Dataset.name=Test000046_-
Dataset.type=Text
Dataset.id=Test000046_-_Z01
Dataset.desc=
Dataset.owner=ugsplm1
Dataset.group=dba
Dataset.created=05-Jan-2010 10:36
Dataset.modified=07-Jan-2010 09:25
Revision.released=
```

ADMIN TOOLS

```
Revision.last_rel_stats=  
NamedRef.created=07-Jan-2010 09:25:01  
NamedRef.modified=14-Mar-2010 15:16:56
```

16.7. Zu Beachten

- Die Ausführung dieses Programms muss als Anwender der Gruppe „dba“ mit Bypass-Recht erfolgen!

17. EXCHANGE - TCPB_EXPORT_VIEW

17.1. Beschreibung

Dieses Programm ist eine Alternative zum Standard PLMXML-Export, wenn mehrere Views unter einer ItemRevision existieren. Das TC Standard Exportwerkzeug unterstützt mehrere Views nicht exakt (d.h. es wird immer der erste View in der Liste exportiert, ohne Rücksicht auf den Typ). TCPB_export_view kann für mehrere Views eines vorgegebenen Typs einen PLMXML-Export initiieren.

17.2. Aufruf

```
tcpb_export_view [-u=<> -p=<> [-g=<>]] -view_type= -transfermode= -xml_file= -rev_rule= -item= -rev= [-help]
```

17.3. Argumente

-u	= Login User
-p	= Login Password
-g	= Login Group
-view_type	= Viewtyp, der exportiert werden soll
-transfermode	= PLMXML-Export transfer mode der benutzt wird
-xml_file	= XML Ausgabedatei (kompletter Pfad)
-rev_rule	= Gültige Konfigurationsregel (Revisionrule) für die Abarbeitung der Views
-item	= Item ID
-rev	= Revision ID
-h	= Ausgabe der Hilfe

17.4. Umgebung

Zum Start des Programms wird eine TC-Command Shell benötigt.

17.5. Dateien

Es werden die Standard TC und PLM-Easy Logfiles angelegt.

18. EXCHANGE - TCPB_IMPORT_FILE

18.1. Beschreibung

Das Programm ermöglicht die Anlage von Datasets und den Import der zugehörigen Dateien. Optionen erlauben eine gezielte Auswahl der Dataset-Typen , der Referenz und eine flexible Steuerung des Imports.

18.2. Aufruf

```
tcpb_import_file -u= -g= -p= [-i=|-f=] -d= -type= -ref= -item= [-rev=] [-rel=] [-de=] [-owner=] [-group=] [-status=]
```

18.3. Argumente

```
-u      = Login User
-p      = Login Password
-g      = Gruppe
-i      = Eingabedatei für Mehrfachimport
-f      = Importdatei, Einzelimport
```

Hinweis:

Entweder die Option `-i` oder `-f` muss gesetzt sein. Wenn die Option `-i` gewählt wird, so enthält die <Eingabedatei> die <Importdatei> mit den folgenden Parametern, andernfalls sind die Daten in der Befehlszeile anzugeben.

```
-d      = Name des Datasets
-type   = Dataset Typ - e.g. UGPART,TIFF ...
-ref    = Named Reference
-item   = Item Id           - wird neu erzeugt, falls nicht vorhanden
-rev    = Revision Id       - wird neu erzeugt, falls nicht vorhanden
                                wenn -rev nicht angegeben ist, wird der Standard verwendet.
-rel    = Relation          - Kompletter Name der Relation, die zwischen Item und ItemRevision besteht.
                                Diese ist nur bei der Neuanlage des Datasets erforderlich.
                                Default ist IMAN_specification.
```

ADMIN TOOLS

-owner = Besitzer des neuen Datasets oder IREV
-group = Gruppe des neuen Datasets oder IREV
-status= Dieser Status wird auf das Dataset gesetzt oder IREV

Hinweis:

Wenn Besitzer/Gruppe oder Status von der ItemRevision bezogen werden soll, so ist das Keyword *IREV* zuzuweisen.

-de = [n|r|c|a] - definiert eine Aktion
 n = new - neuer Dataset wird erzeugt
 r = replace - NR wird ausgetauscht, falls der Dateiname identisch ist,
 andernfalls wird die Datei hinzugefügt
 c = change - NR wird ausgetauscht, dies geschieht unabhängig vom Dateinamen
 a = add - NR wird ohne Prüfung an das Dataset angehängt

Wenn -de nicht definiert wird, erfolgt eine Überschreibung der bestehenden Named Reference gleichen Typs.

-h[elp] = Ausgabe der Hilfe

18.4. Umgebung

Zum Start des Programms wird eine TC-Command Shell benötigt.

Siehe auch Teamcenter Help ⇒ Configuring Teamcenter ⇒ Preferences and Environment Variables Reference.

18.5. Dateien

Es werden die Standard TC Log files angelegt, siehe auch Teamcenter Help ⇒ Administering Teamcenter ⇒ Utilities Reference ⇒ Getting started ⇒ Log files.

18.6. Beispiel

Es wird ein PDF Dokument unter das Item 00001, ItemRevision 01 als Dataset PDF mit dem Namen Schulung und der Benannten Referenz (Named Reference) PDF erzeugt. Besitzer und Gruppe entsprechen der ItemRevision.

```
tcpb_import_file -u=infodba -p=<password> -g=dba -f=test.pdf -d=Schulung -type=PDF -ref=PDF  
-item=000001 -rev=01 -owner=IREV -group=IREV -status=IREV -de=n
```

18.7. Zu Beachten

Die Ausführung dieses Programms erfordert DBA Rechte!

19. EXCHANGE - TCPB_SYNC_CADATTRIBUTES

19.1. Beschreibung

Dieses Programm dient zum Export und Import von Attributen, nach der Definition im CAD Attribute Mapping.

19.2. Aufruf

```
tcpb_sync_cadattributes -u= -g= -p= [-i= | [-f=] [-export|-import] -d= -type= -item= [-rev=] [-rel=]]
```

19.3. Argumente

-u	= Login User
-p	= Login Passwort
-g	= Login Gruppe
-i	= Eingabedatei für mehrfach Im- oder Export Diese Eingabedatei enthält dann die unten angegebenen Parameter. Diese Parameter können auch für einzelne Im- oder Exports in der Kommandozeile angegeben werden, dann ist aber der Parameter -i nicht zulässig.
-f	= Name der Attributedatei - <Eingabedatei für Mehrfachimport> Default: "<dataset name>.txt"
-export	Export der Attribute in ein Datei (default)
-import	Import der Attribute aus einer Datei
-item	= Item Id
-rev	= ItemRevision Id - wird diese nicht angegeben, so wird automatisch die letzte ItemRevision verwendet.
-rel	= TC Relation zwischen Item und Dataset (default: IMAN_specification)
-type	= Datensatz Typ
-d	= Datensatz Name
-both	Ex-/Importiere Attribute mit /master=both
-iman	Ex-/Importiere Attribute mit /master=iman
-cad	Ex-/Importiere Attribute mit /master=cad
	Mappings mit undefiniertem master werden wie /master=iman behandelt
	Default Parameter sind
	Für den Export: -both -iman -cad
	Für den Import: -both -cad
-h -help	Ausgabe der Hilfe.

19.4. Umgebung

Zum Start des Programms wird eine TC-Command Shell benötigt. Siehe auch Teamcenter Help ⇒ Configuring Teamcenter ⇒ Preferences and Environment Variables Reference.

19.5. Zu Beachten

Die Parameter -u, -p und -g müssen immer zusammen oder überhaupt nicht (autologin) angegeben werden.
Die Parameter „-i“ und „-f“ schließen sich gegenseitig aus.

19.6. Beispiele

19.6.1. Beispiel: Export von Attributen

Kommando

```
tcpb_sync_cadattributes -u=infodba -p=<password> -g=dba -f=000039_-Z01.txt -export -item=000039 -rev=1  
-rel=IMAN_specification -type=UGPART -d=000039_1_Z01 -both -iman -cad
```

19.6.2. Format der Ausgabedatei:

Die Syntax ist immer in der Ersten Zeile beschrieben: Name;Type;Master;WriteOnce;Description;Value

Name	Attributname nach CAD Attributemapping
Type	Attribute Type
Master	Master nach CAD Attributemapping
WriteOnce	Parameter aus CAD Attributemapping
Description	Beschreibung nach CAD Attributemapping
Value	Attribut Wert

Beispieldatei:

```
Name;Type;Master;WriteOnce;Description;Value  
DB_PART_TYPE;String;none;false;Part Type;Item  
DB_PART_REV;String;iman;false;Part Revision;1  
DB_PART_NO;String;iman;false;Part Number;000039  
DB_PART_NAME;String;iman;false;Part Name;test  
DB_PART_DESC;String;iman;false;Part Description;  
DB_UNITS;String;iman;false;Part Unit of Measure;
```

Das Format der Inputdatei beim Import von Attributen entspricht dem der Outputdatei.

CREATEIMAGE SERVER FUNKTIONEN

20. CRI - TCPB_COUNT_DATASETS

20.1. Beschreibung

Zählt nach einer konfigurierbaren Syntax die Anzahl der Datasets, die in einer Revision enthalten sind.

Diese Programm steht nur für den CreateImageServer zur Verfügung!

21. CRI - TCPB_IDEAS_USER_ATTRIBUTE

21.1. Beschreibung

Exportiert Attribute von Ideas-Datasets aus TC.

Diese Programm steht nur für den CreateImageServer zur Verfügung!

22. CRI - TCPB_JOB_UTIL

22.1. Beschreibung

Exportiert JobItems aus TC und wandelt diese in Job-Aufträge für den CreateImageServer um.

Diese Programm steht nur für den CreateImageServer zur Verfügung!

23. CRI - TCPB_SET_DECISION

23.1. Beschreibung

Dieses Programm erzwingt im Workflow einen Taskstate Wechsel, indem die Perform Task mit einem Entscheidungsvorgang approve/reject/no abgeschlossen wird. Da dieses Programm interne Jobinformationen benötigt, wird die Anwendung dieses Programms nur in Verbindung mit anderen Programmen funktionieren.

23.2. Aufruf

```
tcpb_set_decision.exe -u=<user> -p=<password> -g=<group> -j=<jobtag_file> -d=[approve|reject|no|delete]  
-c=<comment> -l=[<level_name>|any] -help
```

23.3. Argumente

```
-u      = Login User  
-p      = Login Password  
-g      = Login Gruppe  
-j      = Eingabedatei, die Job-ID enthält  
-d      = Entscheidung <approve=genehmigen|reject=ablehnen|no=keine>  
-c      = Kommentar der Entscheidung  
-l      = Levelname der Prozessstufe oder any  
-help   = Ausgabe der Hilfe
```

23.4. Umgebung

Zum Start des Programms wird eine TC-Command Shell benötigt.

Siehe auch Teamcenter Help ⇒ Configuring Teamcenter ⇒ Preferences and Environment Variables Reference.

23.5. Dateien

Es werden die Standard TC Log files angelegt, siehe auch Teamcenter Help ⇒ Administering Teamcenter ⇒ Utilities Reference ⇒ Getting started ⇒ Log files.

23.6. Beispiel

```
tcpb_set_decision.exe -u= -p= -g= -j=D:\ugsshare\plm0local\CreateImage\job_queue\20040719163545infodba\jobtag.txt  
-d=delete -c=comment -l=any
```

23.7. Zu Beachten

- Die Ausführung dieses Programms ist als Single Prozess nur möglich, wenn die Job-ID in einer Eingabedatei vorliegt. Die Datei ist derzeit nur in Verbindung mit der Create Image-Lösung innerhalb Teamcenter verfügbar. Die erforderliche Job-ID wird in der Create Image-Lösung automatisch erzeugt und gelesen.
- Es wird geprüft, ob der Job referenziert ist. Sollte er referenziert sein, wird er nicht gelöscht. Der Anwender wird informiert, wo der Job noch referenziert ist.
- Die Parameter -u, -p und -g müssen immer zusammen angegeben werden.
- Rechte auf den Job müssen durch Einstellung im Rule-Tree vorhanden sein, damit der Job gelöscht werden darf.