

HBB LayerManager

Version: v1.1.10.0

Datum: 19.03.2019

Ersteller: Karl Wenz

1 Was macht das Programm?

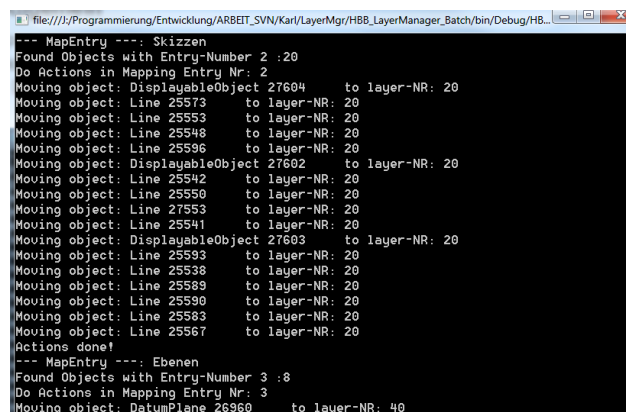
Mit dem Programm „*HBB LayerManager*“ lässt sich eine automatische Layerzuordnung von Objekten durchführen. Die Objekte (Körper, Kurven, Skizzen, Bezugselemente, ...) in einer NX-Datei werden dabei anhand verschiedener Kriterien erkannt und auf einen entsprechenden Layer verschoben.

Einige Kriterien sind z.B.:

- Objekt-Typ anhand von Typbezeichnung oder Typnummern
- Objektname
- Objektfarbe
- Feature-Typ

Zusätzlich können mit diesem Programm auch vordefinierte *Layerkategorien* erstellt werden. Für diese kann ein Name und die zugehörigen Layernummern in einer Textdatei angegeben werden.

Das Programm kann innerhalb von NX und außerhalb über ein Skript aufgerufen werden.



```
file:///J:/Programmierung/Entwicklung/ARBET_SVN/Karl/LayerMgr/HBB_LayerManager_Batch/bin/Debug/HB...
--- MapEntry ---: Skizzen
Found Objects with Entry-Number 2 :20
Do Actions in Mapping Entry Nr: 2
Moving object: DisplayableObject 27604 to layer-NR: 20
Moving object: Line 25573 to layer-NR: 20
Moving object: Line 25553 to layer-NR: 20
Moving object: Line 25548 to layer-NR: 20
Moving object: Line 25596 to layer-NR: 20
Moving object: DisplayableObject 27602 to layer-NR: 20
Moving object: Line 25542 to layer-NR: 20
Moving object: Line 25550 to layer-NR: 20
Moving object: Line 27553 to layer-NR: 20
Moving object: Line 25541 to layer-NR: 20
Moving object: DisplayableObject 27603 to layer-NR: 20
Moving object: Line 25533 to layer-NR: 20
Moving object: Line 25538 to layer-NR: 20
Moving object: Line 25589 to layer-NR: 20
Moving object: Line 25590 to layer-NR: 20
Moving object: Line 25583 to layer-NR: 20
Moving object: Line 25567 to layer-NR: 20
Actions done!
--- MapEntry ---: Ebenen
Found Objects with Entry-Number 3 :8
Do Actions in Mapping Entry Nr: 3
Moving object: DatumPlane 26960 to layer-NR: 40
```

1.1 Systemvoraussetzungen

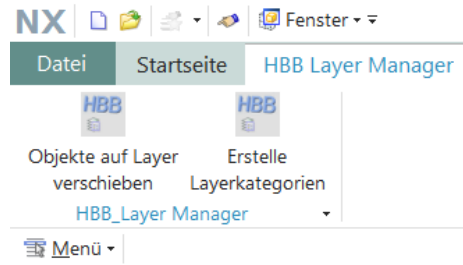
NX-Version: NX10.0

Betriebssystem: Windows (64Bit)

2 Programmaufruf

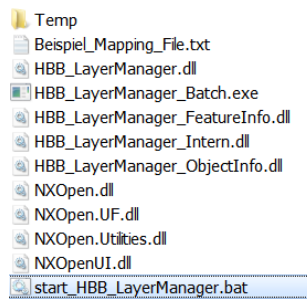
2.1 Aufruf des Programms in NX

Die Funktionen des Programms in NX direkt über den Reiter HBB Layer Manager aufgerufen werden. Dabei wird entsprechend nur das aktuelle Part behandelt.



2.2 Externer Aufruf des Programms

Das Programm kann auch über ein Skript im Programmordner direkt gestartet werden. Dabei werden alle notwendigen Parameter direkt im Skript (*.bat) angegeben.



Auszug aus dem mitgelieferten Beispiel Skript:

Die möglichen Parameter werden auf der nächsten Seite beschrieben!

```
set UGII_BASE_DIR=C:\Siemens\NX10.0
set PATH=%UGII_BASE_DIR%\UGII;%PATH%

REM Temp-Verzeichnis von NX setzen, damit Log-Files dort landen
set UGII_TMP_DIR=%CD%\Temp
if not exist "%UGII_TMP_DIR%" mkdir "%UGII_TMP_DIR%"

REM NX-Logfiles erhalten nach dem der Prozess gestartet wurde
set UGII_KEEP_SYSTEM_LOG=yes

REM PARAMETER für Part-Verzeichnis
Set PART_DIR=C:\training\Test_Layer_Rheinmetall

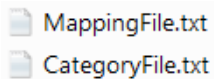
REM Programmaufruf
start "" "%CD%\HBB_LayerManager_Batch.exe" -MappingFile:"%CD%\Beispiel_Mapping_File.txt" -PartDirectory:%PART_DIR% -noDryRun
```

2.3 Parameter für den Programmaufruf

Mit folgenden Parametern kann das Programmverhalten gesteuert werden.

Name	Nutzen	Relevanz
-noDryRun	Speichert die Teile nach Programmablauf	Optional
-Log:<Pfad Logdatei>	Optionaler Parameter für externes Logfile	Optional
-DeleteCategories	Löscht alle existierende Layerkategorien	Optional
-PartDirectory:<Pfad Teileverzeichnis>	Programm durchläuft alle Teile im Verzeichnis	Benötigt für externe Programmversion
-Categories:<Pfad zur Kategoriedatei>	Gibt den Pfad zur Layerkategorie-definitionsdatei an	Benötigt für beide Programmversionen
-MappingFile:<Pfad zur Mappingdatei>	Gibt den Pfad zur Mappingdatei an	Benötigt für beide Programmversionen

3 Konfiguration



Der *HBB LayerManager* verfügt über zwei Konfigurationsdateien welche standardmäßig im Installationsverzeichnis im Unterordner „bin“ zu finden sind.

CategoryFile.txt

In der „*CategoryFile*“ können Layerkategorien definiert werden welche beim Programmstart in das Part übernommen werden.

```

1 NoOutput
2 DOKUMENTATION| Dokumentation | 120-129
3 ELEKTRIK | | 150-169
4 FINITE-ELEMENTE | | 140-149
5 MODEL | | 1-69; 80-99
6 M_BEZUGSOBJEKTE_EBENEN-ACHSEN | | 40-59
7 M_BLECHABWICKLUNG | | 80-89
8 M_HILFSGEOMETRIE | | 60-69
9 M_NORM-KAUFTEILE | | 10-14
10 M_SKIZZEN | | 20-39
11 M_SOLID | | 1-9
12 M_SOLIDS | | 1
13 M_WAVE-GEOMETRIE | | 90-99
14 UNIGRAPHICS | Reserved to UG | 251-256
15 ZEICHNUNG | | 70-74; 100-115; 130
16 Z_2D_ALLGEMEIN| Bemassung_Symbole_usw | 110
17 Z_3D_BEMASSUNG-TEXTE | | 70
18 Z_AUFKLEBER | | 130
19 Z_RAHMEN | | 100
20 Z_TEXTE-DEUTSCH | | 101
21 Z_TEXTE-ENGLISCH| | 102
22 Z_TEXTE-FRANZ | | 103
23 Z_TEXTE-SPANISCH | | 104

```

Beispiel-Eintrag:

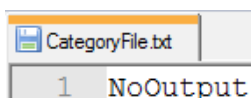
Z_2D_ALLGEMEIN| Bemassung_Symbole_usw | 110; 221-244

Rot: Der Rot umrandete Bereich legt den Namen der Layerkategorie fest

Blau: Der grün umrandete Bereich legt die Kategorie-Beschreibung fest

Grün: Der blau umrandete Bereich definiert welche Layer Teil dieser Layerkategorie sind

Die Drei verschiedenen Bereiche werden durch ein „|“ getrennt. Das Zeichen kann über die Tastenkombination ALTGr + <> eingefügt werden.



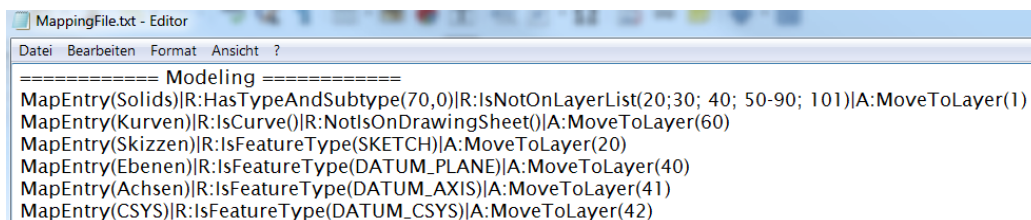
Zusätzlich kann durch das schreiben von „NoOuput“ in eine Zeile die Ausgabe bei der Erstellung von Layerkategorien abgeschaltet werden

Zeichnet eine Kommentarzeile aus.

MappingFile.txt:

In der Datei *MappingFile.txt* wird festgelegt welche Objekte auf welchen Layer verschoben werden sollen. Jede Zeile definiert einen Eintrag.

Dabei kann es relevant sein, in welcher Reihenfolge man Einträge anordnet. Als Beispiel dienen die Typen Skizze und Kurven. Da die Skizze ebenfalls Kurven enthält, sollte man zuerst die „einfachen Kurven“ als Eintrag anlegen und danach dann einen Eintrag für den Typ Skizze.



```

===== Modeling =====
MapEntry(Solids)|R:HasTypeAndSubtype(70,0)|R:IsNotOnLayerList(20;30; 40; 50-90; 101)|A:MoveToLayer(1)
MapEntry(Kurven)|R:IsCurve()|R:NotIsOnDrawingSheet()|A:MoveToLayer(60)
MapEntry(Skizzen)|R:IsFeatureType(SKETCH)|A:MoveToLayer(20)
MapEntry(Ebenen)|R:IsFeatureType(DATUM_PLANE)|A:MoveToLayer(40)
MapEntry(Achsen)|R:IsFeatureType(DATUM_AXIS)|A:MoveToLayer(41)
MapEntry(CSYS)|R:IsFeatureType(DATUM_CSYS)|A:MoveToLayer(42)

```

Eine Zeile besteht aus beliebig vielen Abschnitten die durch das Zeichen „|“ getrennt sind. Das Zeichen kann über die Tastenkombination ALTGr + <> eingefügt werden.

Zeichnet eine Kommentarzeile aus.

R: Definiert eine Erkennungsregel. R steht für Rule.

A: Steht für eine Aktion, welche für erkannte Objekte ausgeführt wird.

Das A steht für Action.

Beispiel-Eintrag:

```
MapEntry(Solids)|R:HasTypeAndSubtype(70,0)|R:IsNotOnLayerList(20;30; 40; 50-90; 101)|A:MoveToLayer(1)
```

Rot: Der rot umrandete Bereich kennzeichnet den Beginn eines neuen Mappingeintrages. Die in den Klammern stehende Bezeichnung dient zur Übersichtlichkeit (Eintragsbezeichner). Der Name wird bei der Ausführung des Programms angezeigt.

Grün: Im Grün umrandeten Bereich befinden sich alle „Rules“ auf welche die Objekte geprüft werden. Eine Liste möglicher Regeln finden Sie unten in der Tabelle.

Blau: Im blau umrandeten Bereich werden Aktionen angegeben. Die Aktionen werden ausgeführt wenn ein Objekt alle „Rules“ (Bedingungen) positiv erfüllt. Als Aktion steht momentan nur „MoveToLayer“ zur Verfügung!

3.1 Liste mit Erkennungsregeln/Aktionen für Objekte

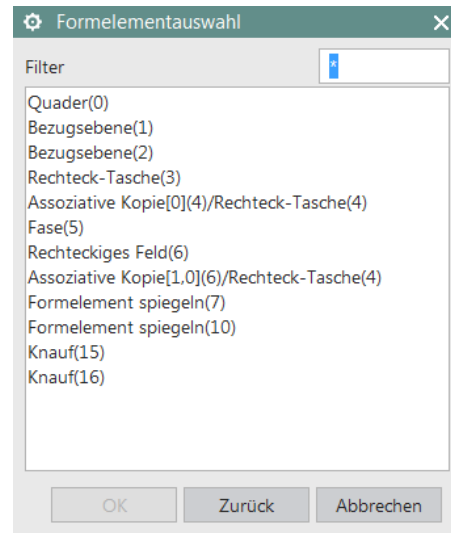
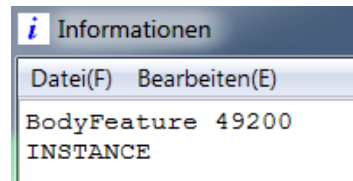
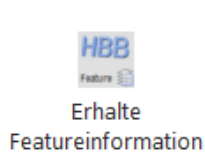
Regelname	Bedingung Prüft ob das Objekt...
R:IsWaveGeo()	...Wavegeometrie ist
R:HasObjectColor(<i>Farbnummer</i>)	...die angegebene Farbnummer hat
R:HasObjectName(<i>Objektname</i>)	...den angegebenen Namen trägt
R:HasTypeAndSubtype(<i>Typ, Subtyp</i>)	...die angegebene Typ-Nr. und Subtyp-Nr. hat
R:IsType(<i>Typname</i>)	...den angegebenen Typnamen hat
R:IsRoutingObject()	...ein Routing-Objekt ist
R:IsSplineSegmentPoint()	...ein Punkt aus einem zugehörigen Spline-Segment ist (Routing!)
R:IsFeatureType(<i>Featuretype</i>)	...vom angegebenen Featuretyp ist
R:IsNotOnLayerList(<i>Layer-Nummern</i>)	...nicht auf einem der angegebenen Layern ist (Trennzeichen ; zwischen den Nummern)
R:IsOnLayer(<i>Layer-Nummer</i>)	...auf dem angegebenen Layer ist
R:IsCurve()	...eine Kurve ist
R:IsOnModelingView()	...in der Modellansicht ist (z.B. bei PMI)
R:IsOnDrawingSheet()	...auf dem Zeichnungsblatt liegt (z.B. bei Rahmen, Maßen)
R:NotIsOnDrawingSheet()	... nicht auf dem Zeichnungsblatt liegt
A:MoveToLayer(LayerNr.)	Verschiebt die identifizierten Objekte auf den angegebenen Layer

4 Analyse von bestehenden Objekten

Um bestimmte Objektarten eindeutig identifizieren zu können, wurden dem Programm noch 2 Analysetools beigelegt (HBB FeatureInfo & HBB ObjectInfo)

4.1 HBB FeatureInfo

Mit dem Button „Erhalte Featureinformation“ lassen sich schnell Featureinformationen von den im aktiven Teil enthaltenen Features abfragen. (Featuretype und Objektbezeichner)



4.2 HBB ObjectInfo

Mit dem Button „Erhalte Objektinformationen“ lassen sich Objektinformationen wie der Typ und Subtyp, sowie auch der Objektname und der Namensraum des Objektes herausfinden.

