

471627

Datenmigration in den Vault

Markus Dössinger
coolOrange s.r.l.

Lernziele

- Phasen und Prozesse der Migration kennenlernen und verstehen
- Was ist das Autodesk Data Transfer Utility und wie benutzt man es?
- Werkzeuge für die Manipulation und Validierung kennenlernen
- Beispiele für erfolgreiche Projekte

Beschreibung

Sie wollen Daten und Dateien aus einem bestehenden Vault oder einem anderen Datenmanagement-System in einen neuen Vault übertragen? Der Vortrag gibt einen Überblick über die Werkzeuge, Phasen und Prozesse für eine Datenmigration nach Vault. Hierbei wird sowohl das Autodesk Data Transfer Utility für den Ex- und Import, sowie Lösungen für die Manipulation und Aufbereitung der Daten vorgestellt.

Redner



Markus Dössinger ist Mitbegründer und Berater bei coolOrange. Er führt Teams durch Autodesk-Datenmanagement-Projekte mit Schwerpunkt auf der Migration zu Vault. In diesem Zusammenhang arbeitet Markus mit dem Autodesk Support an Migrationen von Autodesk Productstream Professional nach Autodesk Vault Professional oder Workgroup. (Zuvor arbeitete Markus als Produktdesigner für Datenmanagement-Produkte bei Autodesk.)

Phasen und Prozesse der Migration

Um Daten und Dateien aus einem bestehenden Vault oder einem anderen Datenmanagement-System in einen neuen Vault zu übertragen, sollte man sich die Phasen und Prozesse für solch eine Datenmigration ansehen.

Phasen der Migration

Die Migration unterteilt sich in 4 verschiedene Phasen.

Analyse

Die erste Phase ist dabei die Analyse. Sie gilt der Bestimmung der Ausgangslage und Datenqualität und Datenquantität.

Ist es schwierig die Daten innerhalb des PDM-Systems zu analysieren, ist es hilfreich die Daten aus dem System zu exportieren und die Analyse auf den Export zu machen. Ein Export ist auch für die spätere Modifikation notwendig und dabei fällt auch die Bestimmung der Export-Laufzeiten als Nebenprodukt ab.

Zur Analyse gehört auch die Prüfung der CAD-Dateien, ob z.B. alle Referenzen gefunden werden. Oft ist diese Analyse im Quellsystem nicht machbar und es muss über Alternativen nachgedacht werden. Ein Weg könnte die Prüfung nur auf benannte Referenzbaugruppen und -Zeichnungen sein. Damit kann man die Dateiqualität zumindest für einen Teil feststellen.

Definition

Die Definitionsphase ist oft die längste und auch schwierigste Phase, da hier viele Diskussionen mit allen Beteiligten geführt werden müssen. Sie ist jedoch auch eine sehr wichtige Phase, die über den Erfolg und Misserfolg der Migration entscheidet. Der Kunde muss definieren, welche Objekte überhaupt in den Vault übernommen werden sollen. Es gilt einige Fragen zu klären.

Sollen nur Dateien oder auch Artikel und Stücklisten übernommen werden?

Sollen wirklich alle Dateien übernommen werden? Kann man sich von Altlasten trennen? Macht es Sinn nur aktuelle Konstruktionen in den Produktiv-Vault zu übernehmen und die älteren Konstruktionen in einen Archiv-Vault abzulegen?

Gibt es weitere Objekte, aus dem Quell-System, die im Ziel-Vault benötigt werden?

Wenn ja, wie werden sie im Vault dargestellt? Wenn es keine Standard-Objekte dafür im Vault gibt, kann man diese als Custom-Objects übernehmen? Wie ist dann die Arbeitsweise im Vault? Welche Anpassungen sind dafür nötig?

Der Kunde muss sich aber auch mit dem Vault als Ziel-System beschäftigen, und zusammen mit dem Autodesk-Partner den Ziel-Vault definieren.

Bewährt hat sich hierbei, typische Konstruktionen als Referenz-Baugruppen und -Zeichnungen zu benennen. An diesen kann man bestimmen und später auch prüfen, wie sie im Vault ankommen sollen. Hierbei können die Zuordnungen von Lebenszyklen, Kategorien und Eigenschaften exemplarisch besprochen und definiert werden.

Insbesondere bei größeren Projekten ist die Definitionsphase eine sich wiederholende Phase. Vor allem die Zuordnung der Eigenschaften bedarf der Anpassung nach den Testläufen. Auch hier bewähren sich wieder die Referenz-Baugruppen.

Realisierung

Die Realisierung ist stark abhängig von der vorherigen Definitionsphase. Zu ihr gehört die Extraktion der Daten und Dateien, sowie die Modifikation der exportierten Daten gemäß den definierten Zuordnungen auf den Ziel-Vault.

Danach erfolgt die Erzeugung eines BCP-Pakets und die Validierung dieses BCP-Pakets. Dies ist ein iterativer Prozess und es muss ein Testsystem zur Verfügung stehen, das dem Ziel-System möglichst nahekommt und mit dem die Testimporte in den Vault durchgeführt werden können. Über die Sichtung des Test-Vault lässt sich entscheiden, ob die Realisierung des BCP-Pakets oder die Definition angepasst werden muss.

Dies ist die Stufe, die im weiteren Verlauf der Präsentation genauer betrachtet wird.

Import

Vor dem Import in das Produktiv-System muss der Import in das Testsystem mit einem Abnahmeprotokoll abgenommen werden. Ab diesem Zeitpunkt dürfen keine Änderungen mehr im Prozess, an den Daten und dem System vorgenommen werden. Es gibt 2 Möglichkeiten für die Erstellung des Ziel Vaults. Die erste Möglichkeit ist ein weiterer Import in den Produktiv-Vault. Die zweite Möglichkeit besteht darin, aus dem abgenommenen Test-Vault ein Backup und auf dem Produktiv-System die Wiederherstellung durchzuführen.

Die Realisierung – eine Herausforderung

Der herkömmliche Weg die Daten vom Quellsystem an den Vault zu übertragen ist mit Hilfe einer individuell angepassten Softwarelösung.

Diese muss dabei die Daten aus dem Quellsystem extrahieren.

Handelt es sich hierbei um ein fremdes PDM-System, muss sich der Entwickler mit dem System bereits auskennen oder sogar einarbeiten. Selbst wenn die Daten in ein „neutrales“ Format, wie z.B. CSV-Dateien oder Excel-Tabellen exportiert werden können, muss man sich mit der Struktur der Export-Tabellen auseinandersetzen.

Ist die Quelle ein Vault, ist es in der Regel einfacher jemanden im Autodesk-Umfeld zu finden, der die Daten über die Programmierschnittstelle (API) auslesen kann.

Alternativ besteht die Möglichkeit aus dem Vault ein BCP-Paket zu erstellen und dieses als Datenquelle zu verwenden. Das hat den Vorteil, dass die Quelle und das Ziel das gleiche Format haben, da das Tool am Ende auch ein BCP-Paket erzeugen muss. Der Nachteil besteht im zusätzlichen Schritt.

Als nächstes müssen die eingelesenen Daten gemäß der, in der Definitionsphase definierten Regeln, an den Ziel-Vault angepasst werden. Hierzu braucht der Entwickler klar formulierte Regeln und Zuordnungen.

Aus diesen Daten ist dann ein BCP-Paket zu erstellen, das mit dem Data Transfer Utility (DTU) der Autodesk in den Vault importiert werden kann. Damit kann zumindest der Import in den Vault mit Hilfe einer standardisierten Lösung erfolgen.

Wie man leicht sieht, fallen die Schritte Extraktion, Modifikation der Daten, sowie die Erstellung des BCP-Pakets in eine Hand. Damit wird es schwer die Verantwortlichkeiten zu trennen. Dies

ist ein aufwändiger, iterativer und zeitraubender Prozess, der mit erheblichen Kosten und Risiken verbunden ist.

Bevor wir uns die Optimierung der Realisierung ansehen, wollen wir das Data Transfer Utility als zentralen Bestandteil der Migration genauer betrachten.

Dieses wird in jedem Fall für den Import gebraucht. Gegebenenfalls kann es aber auch für einen Export aus einem Vault genutzt werden.

Das Autodesk Data Transfer Utility (DTU)

Was ist das DTU?

Das Vault Data Transfer Utility (DTU), auch als VaultBCP (Vault Bulk Copy Process) bekannt, ist ein von Autodesk entwickeltes Tool, mit dem Daten in großen Mengen und mit hoher Geschwindigkeit aus Autodesk Vault exportiert oder importiert werden können.

Anwendungsbeispiele hierfür sind das Zusammenführen von zwei Vault-Datenbanken, das Bereinigen einer Vault-Datenbank, der Import aus Autodesk Productstream Professional oder der Import aus PDM-Lösungen von Drittanbietern.

Für wen ist das DTU?

Die Autodesk schreibt auf ihrer Knowledge Base: „Bevor Sie das Tool in einer Live-Produktionsumgebung verwenden, muss ein strenges Testprogramm angewendet werden, um sicherzustellen, dass die an die Vault-Datenbank übertragenen Daten korrekt sind und wie erwartet funktionieren. Wenn beim Importieren der Informationen in die Produktions-Vault-Datenbank ein Fehler gemacht wird, können sie nicht rückgängig gemacht werden. Die einzige Möglichkeit, mittels Vault-DTU importierte Daten zu entfernen, besteht darin, die letzte vor der Verwendung erstellte Sicherung wiederherzustellen.

Aufgrund dieser Komplexität und der potenziellen Probleme, die bei falscher Verwendung in einer Produktionsumgebung auftreten könnten, wird das Tool den Kunden nicht zur Verfügung gestellt und sollte nicht als Vorteil eines Vault-Abonnements angesehen werden. Das Vault-DTU-Tool wird nur dem Autodesk-Consulting und zertifizierten Partnern zur Verfügung gestellt...”

Das DTU wendet sich also an Benutzer, die dieses Tool bereits sehr gut kennen und auch umfangreiche Kenntnisse über die Vault-Funktionalität und der Vault-Konfiguration besitzen. Zudem sind Erfahrungen in der Projektabwicklung von Migrationen notwendig. Trotz allem Wissen, sind iterative Testzyklen mit abschließenden Reviews notwendig.

Auch ist ein Verständnis der Struktur des BCP-Pakets und der darin enthaltenen XML-Dateien notwendig. Für die Modifikation des BCP-Pakets sind in der Regel Programmierkenntnisse erforderlich.

Es gibt auch keine Möglichkeit einen Import durch weitere Importe zu korrigieren. Es muss also alles in einem Rutsch importiert werden. Korrekturen können nachträglich in aller Regel nur über die API gemacht werden.

Daher wendet sich das DTU i.d.R. an die erfahrenen Autodesk Partner.

Was kann das DTU?

Das DTU funktioniert mit Vault Workgroup und Vault Professional. Ein Vault Basic muss zuvor auf Vault Workgroup oder Vault Professional hochmigriert werden, bevor das DTU dafür verwendet werden kann.

Es werden folgende Objekte vom DTU exportiert bzw. importiert: Ordner, Dateien, Artikel, Stücklisten, Custom Objects und deren Eigenschaften und Verknüpfungen. Engineering Change Orders werden nicht vom DTU übertragen.

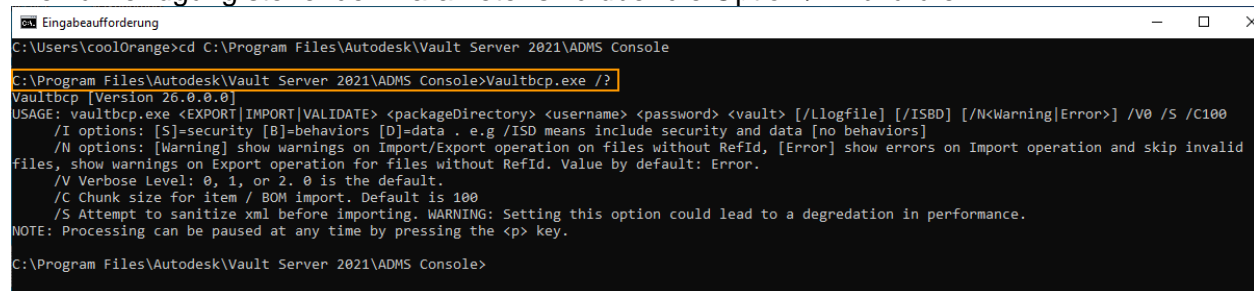
Benutzung des DTUs

Es existiert keine offizielle Dokumentation zum DTU und dem BCP-Format. Auch daher sollte man sich immer an erfahrene Nutzer des DTUs wenden.

Das DTU steht als ZIP-Datei zum Download zur Verfügung. Der Link zum DTU ist bei der Autodesk zu erfragen. Die ZIP-Datei ist in den Ordner der ADMS Konsole zu entpacken.

Das DTU wird über die Datei Vaultbcp.exe angestartet. Dies geschieht über die Eingabeaufforderung oder eine Batchdatei.

Die zu Verfügung stehenden Parameter sind über die Option `/?` Abzurufen:



```

C:\Program Files\Autodesk\Vault Server 2021\ADMS Console>Vaultbcp.exe /?
Vaultbcp [Version 26.0.0.0]
USAGE: vaultbcp.exe <EXPORT|IMPORT|VALIDATE> <packageDirectory> <username> <password> <vault> [/Llogfile] [/ISBD] [/N<Warning|Error>] /V0 /S /C100
/I options: [S]=security [B]=behaviors [D]=data . e.g /ISD means include security and data [no behaviors]
/N options: [Warning] show warnings on Import/Export operation on files without RefId, [Error] show errors on Import operation and skip invalid
files, show warnings on Export operation for files without RefId. Value by default: Error.
/V Verbose Level: 0, 1, or 2. 0 is the default.
/C Chunk size for item / BOM import. Default is 100
/S Attempt to sanitize xml before importing. WARNING: Setting this option could lead to a degradation in performance.
NOTE: Processing can be paused at any time by pressing the <p> key.

C:\Program Files\Autodesk\Vault Server 2021\ADMS Console>
  
```

Zwingend erforderlich ist dabei der Parameter für die Aktion EXPORT, IMPORT oder VALIDATE. Als nächstes ist das „packageDirectory“ anzugeben, das definiert, in welchem Ordner exportiert oder aus welchem Ordner importiert bzw. validiert werden soll. Danach sind der Benutzername, das Passwort und der Vault-Datenbankname anzugeben aus dem exportiert oder in den importiert werden soll.

Es gibt aber auch Optionen, die nicht zwingend erforderlich sind.

Mit der Option `/L` wird der Name der Log-Datei bestimmt.

Mit der Option `/I` lässt sich steuern, ob Security-Objekte, Behavior-Objekte und/oder Daten behandelt werden. Security-Objekte sind z.B. Gruppen, Benutzer und Rollen, die im Vault definiert sind. Beim Import ist diese Option mit großer Vorsicht zu benutzen, da Einstellungen im Ziel-Vault überschrieben werden könnten. Behavior-Objekte sind z.B. die Definition von Kategorien, Eigenschaften, Lebenszyklen oder Revisions-Schemen.

Mit der Option `/NWarning` werden im Log-File Dateien mit fehlender „RefId“ mit einer Warnung angezeigt. Mit der Option `/NError` werden diese Dateien für den Import ignoriert.

Die folgenden Optionen gelten nur für den Import:
 Die Option /V gibt den Grad der Detaillierung für das Logging an.

Die Option /C gibt die Paketgröße an, mit der BOMs importiert werden. So werden in der Voreinstellung 100 Artikel in einem Commit an die Datenbank übergeben. Eine Vergrößerung dieses Wertes bedeutet eine Steigerung der Performance, aber erhöht auch die Anzahl der Artikel, die verloren gehen, wenn beim Import ein Fehler auftritt.

Mit der Option /S wird versucht eine XML vor dem Import zu bereinigen. Diese Option kann aber die Performance verschlechtern.

Ein Beispiel für eine Batchdatei für den Import könnte so aussehen:

```

REM Vault BCP Parameter
SET Vault_Version=Professional 2021
SET ACTION="IMPORT"
SET User="Administrator"
SET Password=""
SET VaultDB="Vault"
SET packageDirectory="C:\temp\VaultBCP\ExportPackages\Export_demo"
SET ISBD=D
REM [S]=security, [B]=behaviours, [D]=data
SET RefId=Warning
SET Verbose=0

REM Run Vaultbcp
"C:\Program Files\Autodesk\Vault Server %Vault_Version%\ADMS Console\Vaultbcp.exe"
%ACTION% %packageDirectory% %User% %Password% %VaultDB%
/L"%packageDirectory%\VaultImportLog.txt" /I%ISBD% /N%RefId% /V%Verbose%
  
```

Der Import kann durch Drücken der p-Taste pausiert werden.

Struktur des BCP-Pakets

Das BCP-Paket besteht aus verschiedenen XML-Dateien.

- **bomBlobs:** Bei Dateien, die mit „bomBlob.“ beginnen, handelt es sich um sogenannte bomBlobs. Diese werden für jede CAD-Datei erstellt und beinhalten Informationen aus der CAD-Stückliste. Da die Dateien meist im weiteren Prozess verändert werden, passen die bomBlobs nicht mehr zu den Dateien. Zudem kostet der Import von bomBlobs viel Zeit. Daher wird empfohlen die bomBlobs für die Migration nicht zu übernehmen, sondern soweit diese im Ziel-Vault benötigt werden, über den Job zu erzeugen, der im Vault zur Verfügung steht. Dieser wird in Vault 2021 mit dem Befehl „Artikeldaten extrahieren“ in die Job Queue gelegt. Zudem werden bomBlobs bei der Bearbeitung und dem Einchecken der CAD-Dateien neu erzeugt.
- Die **BOMwrapper.xml** enthält Stücklisten-Informationen.
- Die **CustomObjectWrapper.xml** enthält die Informationen zu den Custom Objects.
- Die **Itemswrapper.xml** beschreibt die Artikel.
- Die **PropDefswrapper.xml** ist für Definition von iProperties zuständig.
- Die **UOMwrapper.xml** ist für Definition von den Maßeinheiten in Vault zuständig. Diese muss ggf. angepasst werden, wenn die Einheiten von Quell- und Ziel-Vault nicht übereinstimmen.
- **Vault.xml:** Diese XML-Datei enthält Informationen über die Ordner und Dateien und deren Eigenschaften und Verknüpfungen. Auch die Definition der Security-Objekte, wie die Benutzer und Gruppen, sowie die Definition der Funktionsweisen, wie Kategorien, Eigenschaften oder Lifecycles sind hier enthalten.

Zudem gibt es bei einem Export aus dem Vault einen Unterordner für die Dateien, der eine Endung „_1-files-0“ enthält. In diesem Ordner werden alle Dateien abgelegt, die im Exportpaket referenziert werden und bekommen einen fortlaufend durchnummerierten, nicht sprechenden Namen.

Struktur der Vault.xml

Die Struktur der einzelnen XML-Dateien zu erklären, würde hier den Rahmen sprengen. Dennoch möchte ich hier am Beispiel der Vault.xml ein paar grundsätzliche Informationen zur BCP-Struktur geben.

Die Vault.xml beschreibt die Informationen für Ordner und Dateien. Sie ist in mehrere Sektionen unterteilt.

Die erste Sektion ist durch das Element **<Security>** beschrieben. In dieser werden die Gruppen und Benutzer beschrieben.

Die zweite Sektion wird durch das Element **<Behaviors>** beschrieben. Hier finden sich die Definitionen für Kategorien, Eigenschaften, Lebenszyklen und Revisions-Schemen.

Im Element **<Root>**, das die dritte Sektion definiert, werden die Ordner-Struktur und die Dateien, die in dem entsprechenden Ordner liegen beschrieben.

Im Element **<Folder>** werden über Attribute der Ordnername sowie Systemeigenschaften, wie das Erstellungsdatum benannt. Es enthält aber auch Elemente für die Unterordner, die wiederum ein Element **<Folder>** sind, und ein Element **<File>** für jede Datei, die in diesem Folder liegt.

An jedem Element **<File>** gibt es dann die Attribute, die die Datei beschreiben. Wie z.B. der Name oder die Kategorie. Aber es gibt auch Unterelemente für jede Revision. Das Element **<Revision>** wiederum enthält für jede Dateiversion ein Element **<Iteration>**.

Erst in diesem gibt es dann Attribute, die beschreiben, welche Datei aus dem Dateisystem verknüpft ist, wer der Ersteller ist und wann die Version erstellt wurde. Auch werden hier der Status und die benutzerdefinierten Eigenschaften beschrieben. Zudem werden hier die Verknüpfungen zu anderen Dateien über das Element **<Association>** beschrieben.

Wir bekommen also folgendes vereinfachtes Schema:

```

XML-Struktur.xml
1  <Root>
2    <Folder_1>
3      <Folder_2>
4        <File_1>
5          <Revision_A>
6            <Iteration_1>
7              <UDP Name = "UDP1" >"..." </UDP>
8              <UDP Name = "UDP2" >"..." </UDP>
9              <UDP Name = "UDP3" >"..." </UDP>
10             <Association />
11             <Association />
12            </Iteration_1>
13            <Iteration_2>
14              <UDP Name = "UDP1" >"..." </UDP>
15              <UDP Name = "UDP2" >"..." </UDP>
16             <Association />
17            </Iteration_2>
18          </Revision_A>
19          <Revision_B>
20            <Iteration_3>
21              <UDP Name = "UDP1" >"..." </UDP>
22              <UDP Name = "UDP2" >"..." </UDP>
23             <Association />
24            </Iteration 3>
25          </Revision_B>
26        </File_1>
27      </Folder_2>
28    <File_2>
29      <Revision_A>
30        <Iteration_1>
31          <UDP Name = "UDP1" >"..." </UDP>
32          <UDP Name = "UDP2" >"..." </UDP>
33          <Association />
34        </Iteration_1>
35      </Revision_A>
36    </File_2>
37  </Folder_1>
38 </Root>
39

```


Schon in diesem einfachen Beispiel ist zu erkennen, dass die Struktur sehr verschachtelt und nicht einfach zu bearbeiten ist. Auch wenn das Umbenennen einer Datei noch einfach ist, so ist das Verschieben der Datei in einen anderen Folder schon sehr schwierig. Wenn dann auch noch abhängige Dateien mitbehandelt werden sollen, ist eine manuelle Bearbeitung in einem Editor fast unmöglich.

Werkzeuge für die Manipulation und Validierung

Welche Werkzeuge werden im Markt für die Migration angeboten?

Export/Import

Für den Export und Import von Daten und Dateien stellt die Autodesk folgende Werkzeuge zur Verfügung:

Autoloader

Der Autoloader von Autodesk ist nur für die Übernahme von Dateien aus dem Windows Explorer geeignet. Dieses Werkzeug ist geeignet für eine begrenzte Anzahl von Dateien.

Vault API

Für die Verwendung der Vault API sind Programmierkenntnisse notwendig. Bei sehr großen Datenmengen stößt man auch hierbei an seine Grenzen.

Autodesk Data Transfer Utility (DTU)

Das Data Transfer Utility ist das Werkzeug der Wahl, wenn große Datenmengen aus dem Vault exportiert oder in den Vault importiert werden sollen.

Modifikation

Für die Modifikation der Daten werden keine Standardwerkzeuge von der Autodesk angeboten. Es gibt jedoch Werkzeuge, wie zum Beispiel das bcpToolkit von der coolOrange, das als Software Developer Kit die Modifikation und Erzeugung von BCP-Paketen unterstützt. Die Problematik für das Erstellen von Standard-Lösungen für die Modifikation in einem Migrationsprojekt liegt darin, dass die Anforderungen an diese Lösung sehr individuell sind.

Validierung

Für die Validierung eines BCP-Pakets bietet das DTU die Option „VALIDATE“. Hierbei wird ein Import in den Vault simuliert und es werden Warnungen und Fehler angezeigt, die beim Import auftreten würden. Da der Import simuliert wird, ist mit längeren Laufzeiten zu rechnen. Auch hier gibt es im Rahmen des coolOrange bcpToolkit Werkzeuge, die für die Validierung eingesetzt werden können.

An dieser Stelle, möchte ich ihnen ein neues Konzept der coolOrange vorstellen, das eine Vereinfachung für den gesamten Prozess darstellt.

Die Realisierung mit Hilfe einer Zwischendatenbank

Die Idee ist, den Migrationsprozess mit einer neutralen Zwischendatenbank zu standardisieren und zu vereinfachen.

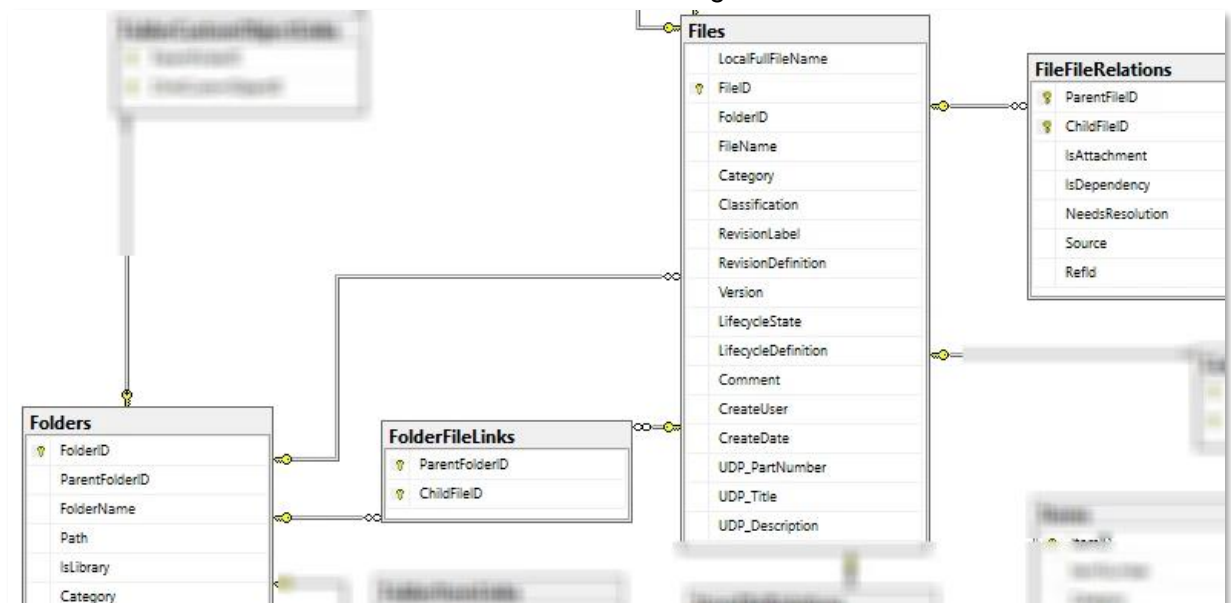
Trennung der Prozesse

Durch die Schaffung einer neutralen Zwischenschicht, lassen sich die notwendigen Schritte trennen und damit auch die Verantwortlichkeiten. Wir benutzen dazu eine SQL-Datenbank, die wir Intermediate Database oder kurz IDB nennen. Diese steht im Zentrum der Migration und ihr Datenbankmodell muss offen und einfach zu verstehen sein.

Damit lässt sich die Bereitstellung der Daten von der Verarbeitung und dem Import trennen.

Datenbankmodell der Intermediate Database

Für die Intermediate Database, die wir in unseren Projekten benutzen, gibt es ein verständliches Datenbankmodell und eine Beschreibung aller Tabellen und Felder.



(Ausschnitt aus Datenbank-Model)

Es wird für die Erzeugung einer solchen Zwischen-Datenbank ein SQL-Skript zur Verfügung gestellt. In der Intermediate Database gibt es für alle Objekte eine eigene Tabelle, in der alle Eigenschaften dieses Objekts enthalten sind. Zum Beispiel sind alle Eigenschaften zu einer Datei, wie die Kategorie, der Lebenszyklus oder benutzerdefinierte Eigenschaften in der Tabelle „Files“ enthalten.

Vorteile der neutralen Zwischen-Datenbank

Durch die klare Trennung von Bereitstellung und Verarbeitung und der Standardisierung des Zwischen-Zielformats, muss eine Partei nur die Quell-Daten zur Verfügung stellen. Durch die Offenlegung und das verständliche Datenbank-Schema ist ihr bekannt, wie sie die Datenbank füllen muss. Die andere Partei, kann sich somit auf die Modifikation der

Daten fokussieren. Da geregelt ist, wie die Daten vorliegen, kann das andere Team die Daten so aufbereiten, dass sie für den Ziel-Vault passen. Jeder ist für seinen Bereich zuständig. Dies bedeutet Klarheit für die Verantwortlichkeiten.

Damit ist es auch möglich eine erste, bereits sehr frühe Validierung in der IDB durchzuführen: Sind alle Daten, die übertragen werden sollen, bereits vorhanden? Liegen die Daten im entsprechenden Format vor? Welche Dateien eignen sich als Referenz-Dateien, um schnelle Zwischenprüfungen durchzuführen?

Wenn z.B. eine Datei am Ende eines Testlaufs nicht im Vault zu finden ist, ist somit sehr leicht zu prüfen, ob sie überhaupt in der zwischen Datenbank vorhanden ist und zur Verfügung gestellt wurde. Ist sie in der zwischen Datenbank nicht zu finden, liegt klar ein Problem in der Bereitstellung vor. Findet man sie jedoch in der Zwischen-Datenbank, weiß man, dass man sich auf dem Prozess der Modifikation oder Erstellung des BCP-Pakets konzentrieren kann. Damit gewinnt man Zeit, da viel schneller klar ist, auf welcher Seite das Problem zu lösen ist. Die Verantwortlichkeit ist damit viel besser geregelt.

Wie die Intermediate Database gefüllt wird, unterscheidet sich vom Anwendungsfall.

Übernahme aus einem Fremdsystem

Das Fremdsystem kann ein PDM-System von einem Drittanbieter oder auch eine selbstentwickelte Datenmanagement-Lösung auf Grundlage einer Excel-Tabelle, einer Access- oder SQL-Datenbank sein. In allen Fällen braucht es ein gutes Wissen über die Quelldatenbank, um die Daten von dort zu extrahieren.

Da das Datenbankmodell der IDB einfach und offen ist, ist die Partei, die die Quelle kennt, in der Lage die Zwischendatenbank z.B. über SQL-Mittel zu füllen.

Die Dateien, die in diesem Prozess involviert sind, werden nicht in die Datenbank übertragen, sondern in einem zugänglichen zentralen Dateisystem zur Verfügung gestellt. In der Intermediate Database steht in einem Feld der Dateiname und Ordner, in dem die Datei gefunden werden kann.

Füllen der IDB von 2 Vault-Datenbanken

Sollen 2 oder mehrere Vault-Datenbanken zusammengeführt werden, ist es naheliegend die BCP-Exportpakete aus den Vaults als Datenquelle zu verwenden. Diese gilt es dann an die Intermediate Database zu übergeben.

Hiermit ist sowohl das Format der Quelle als auch das Format des Ziels definiert. Das erleichtert ein Werkzeug zu schaffen, das das BCP-Paket ausliest und an die IDB übergibt.

Vorteile der Modifikation in der IDB

Die Modifikation in der Datenbank wird mit SQL-Datenbank-Skripten realisiert. Das SQL-Format ist sehr gut dokumentiert, die Ausführung der Skripte ist schnell und reproduzierbar.

Wie bereits erwähnt, gibt es in der Intermediate Database der coolOrange für die zentralen Objekte eine eigene Tabelle, in der alle Eigenschaften dieses Objekts enthalten sind. Möchte man z.B. das Verhalten einer Datei ändern ist nur die Tabelle „Files“ anzusprechen. Würde das Format des BCP-Pakets als Datenbank-Modell

vorliegen, müssten bei Änderungen oft 2 oder sogar 3 Tabellen angesprochen und berücksichtigt werden.

Über die SQL-Mittel lassen sich hilfreiche Analysen durchführen. Z.B. kann geprüft werden, wie oft bestimmte Eigenschaften überhaupt verwendet werden. Man kann untersuchen, ob es überhaupt Sinn macht, diese Eigenschaft in den Vault zu übernehmen oder ob diese gegebenenfalls mit anderen Eigenschaften zu vereinigen sind.

Nach der Modifikation können die Daten z.B. auf Gültigkeit geprüft werden. Sind z.B. alle Mussfelder gefüllt? Somit ist eine frühzeitige Validierung noch vor dem Erstellen des BCP-Pakets möglich.

Erzeugung des BCP-Pakets

Das Datenbank-Model der Intermediate Database ist auf einfache Bearbeitung ausgerichtet, entspricht aber nicht der Struktur eines BCP-Pakets. Daher braucht es auch ein Werkzeug, das aus den Daten in der IDB ein BCP-Paket erstellt.

Validierung

Für die Validierung des BCP-Pakets steht die Option „VALIDATION“ im DTU zur Verfügung. Diese hat aber den Nachteil, dass der Import nachgebildet wird und somit bei großen Paketen auch mit langen Laufzeiten zu rechnen ist.

Daher wäre es wünschenswert, wenn es noch weitere Tools geben würde, die das BCP-Paket auf Gültigkeit prüfen könnten und die Datei- und Ordnerstruktur, wie sie später im Vault vorliegt, schon vor einem zeitraubenden Import für eine erste Prüfung anzeigen könnte.

Import

Das geprüfte BCP-Paket wird mit dem Data Transfer Utility in den Vault importiert. Abhängig von der Datenmenge, kann dies mehrere Tage dauern. Wie bereits erwähnt, lassen sich importierte Daten nicht mit einem weiteren Import korrigieren. Daher sind mehrere Testläufe in einen Ziel-Vault nötig, bis ein finaler Import gefahren werden kann.

Review/Abnahme

Ist der Import abgeschlossen kann ein Review der Logdateien und des Vaults stattfinden. Dazu gehört die Sichtung der Log-Dateien aus dem Import und natürlich eine Sichtung des Vaults selbst. Neben einer stichpunktartigen, zufälligen Prüfung muss eine systematische Prüfung stattfinden. Dies wird wesentlich durch einen Testplan erleichtert, der auf Basis der vorher erwähnten Referenz-Baugruppen, -Zeichnungen und anderer Objekte erstellt wurde.

In einem Abnahmeprotokoll muss festgehalten werden, was korrekt umgesetzt wurde und was verbessert werden muss.

Hieraus ergeben sich dann die Aktionen für den nächsten Testlauf. Die Schritte vom Export bis zum Import müssen solange wiederholt werden, bis ein ausreichendes Ergebnis erreicht ist.

Finaler Import in den Produktiv-Vault

Als letzter Schritt fehlt noch der finale Import in den Produktiv-Vault. Wurde der letzte Testlauf erfolgreich abgenommen, bieten sich 2 Möglichkeiten für den letzten Import in das Produktiv-System an.

Zum einen kann man am Testsystem ein Backup von der abgenommenen Vault-Datenbank machen und diesen am Produktiv-Server wiederherstellen. Dies setzt aber voraus, dass das Testsystem und das Produktiv-System identisch sind.

Die andere Möglichkeit ist, den Import des letzten BCP-Pakets am Produktiv-System zu wiederholen. Auch hierbei ist es von Vorteil, wenn das Test- und das Produktiv-System identisch sind. Dieses Szenario sollte zuvor auf einen Klon des Produktiv-Vault getestet sein, da man sich nicht darauf verlassen kann, dass sich das Produktiv-System identisch zum Testsystem verhält.

Migration mit Hilfe des bcpToolkit

Komponenten

Das bcpToolkit von der coolOrange stellt viele der benötigten Komponenten bereit, die für die Migration benötigt werden. Das bcpToolkit besteht dabei aus dem bcpDevkit, welches ein Software Developer Kit darstellt, um einfacher eigene Lösungen für die Migration mit dem BCP-Paket zu erstellen.

Ergänzt wird das bcpToolkit durch Code-Beispiele, sogenannte Labs-Tools, die auf unserer GitHub-Seite für unsere Partner zur Verfügung stehen. Diese Labs-Tools haben einen Basis-Leistungsumfang, sind jedoch nicht für alle Anwendungsszenarien ausprogrammiert. Durch die Verfügbarkeit des Quell-Codes können sie an das jeweilige Migrations-Projekt angepasst werden. Der Vorteil liegt darin, dass auf bestehenden Code aufgebaut werden kann und somit der zeitliche Aufwand geringer ist, die benötigten Werkzeuge zu erstellen.

Alle Labs-Tools benötigen die bcpToolkit Lizenz.

Realisierung mit dem bcpToolkit

Mit dem bcpToolkit werden die Tools bereitgestellt, die die Lücke schließen die Intermediate Database zu Füllen und das BCP-Paket zu erstellen.

Mit dem Labs-Tool IDB.Load.BCP kann die Information aus einem BCP-Paket an die IDB übertragen werden.

In der Intermediate Database werden die Daten dann aufbereitet und mit dem Labs-Tool IDB.Translate.BCP das modifizierte BCP-Paket erstellt.

Mit den Tools bcpViewer und bcpValidator lassen sich die BCP-Pakete vor dem Import überprüfen.

Sind nur Dateien an den Vault zu übergeben, lassen sich die Datei-Informationen mit dem Labs-Tool IDB.Load.Files an die Intermediate Database übertragen und dort mit dem Labs-Tool IDB.Analyzer.Inventor und IDB.Analyzer.AutoCAD mit den Datei-Verknüpfungs-Informationen ergänzen.

Anwendungsbeispiel: Bereinigung eines Vaults

Am Anwendungsbeispiel einer Übertragung eines Vaults aus einem englischen Server in einen deutschen Server, lässt sich die Migration mit Hilfe des bcpToolkits darstellen:

1. Export des Vaults aus der englischen Umgebung mit dem DTU
2. Kopieren des BCP-Pakets ohne bomBlobs auf den Arbeitsrechner
3. Import in die Intermediate Database mit dem IDB.Load.BCP
4. Modifikation der Daten auf die deutsche Konfiguration mit SQL-Mitteln
 - a. Bereinigen von Ordnern
 - b. Übertragen von Eigenschaftswerte in eine andere Eigenschaft
 - c. Anpassen der Funktionsweisen auf den deutschen Vault
5. Export der Daten in ein BCP-Paket mit dem IDB.Translate.BCP
6. Validierung mit dem bcpViewer
7. Import in die deutsche Umgebung mit dem DTU

Für wen ist das bcpToolkit gedacht?

Da das bcpToolkit keine fertige Lösung, sondern einen Baukasten gleichzusetzen ist, braucht es Kompetenz in Migrationsfragen und auch Programmier-Knowhow.

In aller Regel werden Migrationsprojekte vom Autodesk Consulting oder Autodesk Partnern gemacht. Ein Migrations-Projekt ist nicht auf die Benutzung des bcpToolkits begrenzt. Es braucht Beratung und Schulung zum Vault. Der Vault muss auf den Kunden-Workflow eingerichtet und konfiguriert werden. Die richtige Benutzung des Data Transfer Utilities muss sichergestellt sein.

In Verbindung mit der Intermediate Database können aber Teile der Migration auch vom Endkunden selbst übernommen werden. Zum Beispiel könnte die Intermediate Database von ihm selbst gefüllt werden, oder auch die Modifikation der SQL-Daten in der IDB, wenn das entsprechende SQL-Knowhow vorhanden ist. Dies kann sinnvoll sein, da der Kunde seine Daten am besten kennt.

Auch gibt es die Möglichkeit, Erweiterungen an den Labs-Tools bei der coolOrange in Auftrag zu geben.

Beispielhaft möchte ich 2 Projekte mit der Intermediate Database vorstellen, die dies nochmals verdeutlichen.

Beispiele erfolgreicher Migrationen

Übernahme aus PDM-System PRO.FILE

Bei der Firma Felss Systems GmbH in Königsbach-Stein sollten Dateien und Artikel aus dem PDM-System PRO.FILE nach Vault übertragen werden, um unternehmensweit in der Konstruktion mit Vault zu arbeiten. Die CIDEON ist der Autodesk-Partner, der die Firma Felss bei der Einführung und Betreuung der PDM- und CAD-Landschaft berät.

Der Import aus PRO.FILE in die Intermediate Database wurde dabei direkt vom Partner CIDEON übernommen. Herr Joachim Meister von der CIDEON hat dies mit Hilfe von SQL-Skripten durchgeführt, da auch PRO.FILE auf einer SQL-Datenbank aufbaut. Das Aufbereiten der Daten in der SQL-Datenbank wurde von Hr. Meister bewerkstelligt. Dabei wurde die CIDEON durch ein Mentoring von der coolOrange unterstützt. Die benötigte Erweiterung des Labs-Tools IDB.Translate.BCP durch die coolOrange wurde in Auftrag gegeben. Die Durchführung für die Erstellung des BCP-Pakets und auch der Import in einen Vault lagen wieder bei der CIDEON. Nach wenigen Test-Importen konnte das Projekt diesen September ohne Fehler beim Import fertiggestellt werden.

Das nächste Projekt mit der Intermediate Database sieht bereits an.

Zitat von Herr Meister, CIDEON: „Ich denke das Konzept mit der load-Datenbank ist genau das, was man für solche Übernahmen brauchen kann.“

Zusammenführen von 2 Vault Datenbanken

Die Firma Metso Outotec hatte die Aufgabe die Dateien aus 2 Vaults zusammenzuführen und in einen Produktiv-Ziel-Vault zu importieren. Zudem sollten CAD-Dateien, die in einem Explorer-Verzeichnis aus dem ERP System abgelegt wurden, zusätzlich importiert werden.

Einer der beiden Vaults war ein Vault Basic und musste erst auf ein Vault Professional angehoben werden. Für die ca. 112.000 Dateien aus dem Vault Basic gab es natürlich keine Zuordnungen zu Kategorien, Lebenszyklen oder Revisions-Schemen. Auch die Funktionsweisen des anderen Vaults mussten erst auf die Konfiguration des Ziel-Vaults angepasst werden. Metso hat dabei Mentoring und Unterstützung bei der coolOrange in Auftrag gegeben.

In Begleitung von coolOrange wurden für die beiden Vaults mit dem DTU die BCP-Pakete erzeugt. Mit Hilfe des Labs-Tools IDB.Load.BCP hat der Projektverantwortliche David Bonanni von Metso die Vault.xml in die Intermediate Database geladen. Die Modifikation auf den Ziel-Vault wurde ausschließlich per SQL-Skripte im SQL Management Studio gemacht. Dabei haben die ausführliche Dokumentation und Beispiele, die im Netz zu finden sind, geholfen alle notwendigen Änderungen per SQL umsetzen können.

Die Daten für 515 Dateien aus SAP wurden mit dem Labs-Tool IDB.Load.Files in die Intermediate Database importiert und angepasst. Mit Hilfe des Tools IDB.Analyzer.Inventor konnten die Referenz-Beziehungen und der Wert der RefId aus den Inventor-Dateien extrahiert und an die IDB übergeben werden. Mit SQL-Skripten wurden bereits etliche Validierungen durchgeführt, um die Konsistenz des Pakets sicherzustellen.

Mit dem Labs-Tool IDB.Translate.BCP wurde das BCP-Paket für die Testimporte erzeugt. Mit dem bcpViewer und bcpValidator wurde das BCP-Paket nochmals gegen den Ziel-Vault geprüft. Das Projekt soll bis Ende des Jahres abgeschlossen werden.

Herr Bonanni sieht den Vorteil der Intermediate Database darin, dass er selbst die Modifikation der Dateien in der SQL-Datenbank machen kann. Somit wird hierfür kein Entwickler gebucht und die Abstimmung mit diesem entfällt. Durch die SQL-Skripte sind die Läufe und Tests sehr leicht wiederholbar. Damit können die Aufgabenstellungen schrittweise gelöst werden.

Herr Bonanni hat bestätigt, dass das Konzept der Intermediate Database für ihn ein großer Vorteil ist.