

CONTENTS

| | | |
|----------|---|---|
| 1 | EINLEITUNG | 8 |
| 1.1 | Hintergrund | 8 |
| 1.2 | Dokument Verwendungszweck | 8 |
| 2 | ARCHITEKTUR / FUNKTIONALITÄTEN / FEATURES | 9 |
| 2.1.1 | ALK Daten im EDBS Format nach Autodesk Topobase 2008 | 9 |
| 2.1.2 | Datenmodels | 11 |
| 2.1.3 | Datenmodellupdate | Fehler! Textmarke nicht definiert. |
| 2.1.3.1 | ALK Datenmodellstruktur | 11 |
| 2.1.3.2 | EDBS Import Metadatenstruktur | 14 |
| 2.1.4 | Batchbetrieb | 15 |
| 2.1.5 | Daten löschen | 15 |
| 2.2 | Integration in Autodesk Topobase 2008 Client | 24 |
| 2.3 | Verarbeitung bundeslandspezifischer Objektkataloge | 19 |
| 2.3.1 | Domain Control Table | 20 |
| 2.3.2 | DCT –Verfahren zum Zuordnen ALK Objekte zum Datenmodell | 23 |
| 2.4 | Verarbeitung von BZSN Daten | 17 |
| 2.4.1 | BSPE | 17 |
| 2.4.2 | FLOE/FEIN | 17 |
| 2.4.3 | BZSN-Datenstruktur | 18 |
| 2.5 | Reporting und Logging | 24 |
| 2.6 | Darstellungsmodelle | 24 |
| 2.7 | Dokumentation | 25 |
| 2.8 | Wartungsvertrag | 25 |
| 3 | DELIVERABLES | 26 |
| 3.1 | Components | 26 |
| 3.2 | Release/Versioning - Roadmap | 26 |
| 3.2.1 | Lake Constance (1.Okt 2007) | 26 |
| 3.2.2 | Lake Constance +1 (17.Dez 2007) | 26 |
| 3.2.3 | Lake Constance +2 | 26 |
| 3.3 | Source Code | 26 |
| 3.4 | Workflow/Uses Cases | Fehler! Textmarke nicht definiert. |

2.4 Verarbeitung von BZSN Daten

Der Topobase EDBS Import bietet die Möglichkeit BZSN Daten einzulesen. Zu jedem importierten ALK Objekt werden die Informationen EDBS Schlüssel, Veränderungsnummer (VN-Nummer) und Veränderungsdatum (VN-Datum) im ALK User verwaltet.

Die Verarbeitung von Differenzdaten erfolgt so dass zuerst alle zu verändernden Objekte gelöscht werden und dann die neu, geänderten Objekte importiert werden.

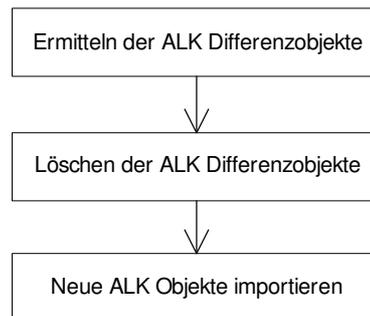


Abbildung 2-4 Verarbeitung Differenzdaten

Die beiden Speicherformate BSPE und FLOE/FEIN werden unterstützt. Eine Unterscheidung bei der Verarbeitung von Differenzdaten vom Typ BSPE oder FLOE/FEIN gibt es nicht. Der Unterschied liegt in der Ermittlung der Differenzdaten.

2.4.1 BSPE

BSPE liefert keinen Löschedatensatz somit muss für jedes zu importierende Objekt geprüft werden ob ein ALK Objekt mit dem EDBS-Schlüssel in der DB schon existiert. Die Entscheidung ob eine Speicherung einer älteren Version gelöscht wird kann nur über den Vergleich der VN-Nummer und des VN-Datums erfolgen. Ist bei gleichem EDBS Schlüssel die VN-Nummer und das VN-Datum beim zu importierenden ALK-Objekt höher liegt ein Differenzfall vor. Das in der Datenbank bestehende Objekt muss zuerst gelöscht werden, bevor das neue ALK-Objekt eingefügt werden kann.

Ist der EDBS Schlüssel nicht in der Datenbank, dann handelt es sich um ein neues Objekt. Das Objekt wird importiert.

2.4.2 FLOE/FEIN

Beim Speicherformat FLOE/FEIN wird eine Änderung/Löschung eines Objektes immer durch das Vorkommen eines FLOE Datensatzes angezeigt. Ein Vergleich mit ALK-Objekten in der Datenbank auf VN-Nummer und VN-Datum beim FLOE/FEIN muss nicht erfolgen.

Der FLOE Datensatz gibt mit Hilfe des EDBS-Schlüssel an welches Objekt gelöscht wird. Der FEIN Datensatz entspricht eine Einfügedatensatz wie BSPE.

FLOE bezieht sich auf Objektebene. Es werden alle mit Hauptobjekt verknüpften Informationen und Geometrien gelöscht. Die Löschung von Teilobjekte wird nicht unterstützt.

Topobase EDBS: Specification

Auch bei FLOE/FEIN werden alle importierten Objekte mit VN-Nummer, VN-Datum und EDBS-Schlüssel in der Datenbank gespeichert.

2.4.3 BZSN-Datenstruktur

Die Verwaltung der EDBS Objekte erfolgt über eine Datenstruktur im Topobase ALK EDBS User. In der Oracle Tabelle (nicht Topobase Featureclasses) EDBS_OBJEKTE werden alle importierten ALK-Objekte mit ihrem:

- EDBS Schlüssel (EDBS_KEY)
- der Dateiimport ID (ID_DATAFILE)
- der FID des erzeugten Topobase Objekts (FID_OBJEKT)
- der Veränderungsnummer (VN_NR)
- dem Veränderungsdatum (VN_DATE)

Informationen zu importierten Dateien werden in der Oracle Tabelle EDBS_DATAFILES gespeichert. Zu jeder Datei wird:

- der Name (NAME)
- der Importpfad der Datei (PATH)
- das Importdatum (IMPORT_DATE)
- der User der den Import durchgeführt hat gespeichert (USER)
- ID-Primary Key (ID)

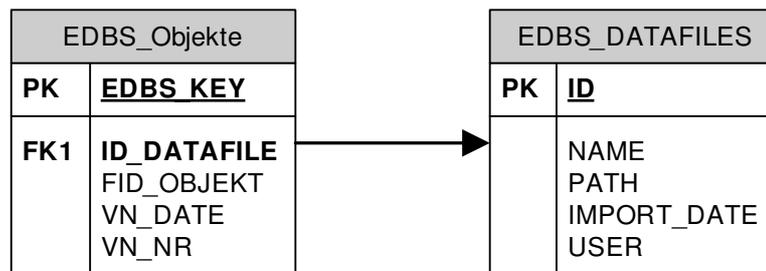


Abbildung 2-5 Datenstruktur BZSN Verfahren

EDBS_OBJEKTE

| Column | Description |
|-------------|--|
| EDBS_KEY | Key 7char, alphanumeric, Datasource distinct key |
| ID_DATAFILE | Foreign Key EDBS_DATAFILES |

| | |
|------------|--|
| FID_OBJEKT | FID Topobase 2008 objects |
| VN_DATE | Date Modified |
| VN_NR | Number Modified, Object Version Number |

Tabelle 2-4 EDBS_OBJEKTE

EDBS DATAFILES

| <i>Column</i> | <i>Description</i> |
|---------------|----------------------------------|
| ID | Primary Key |
| NAME | Name of the imported EDBS File |
| PATH | Importpath of EDBS File |
| USER | OS User who import the EDBS File |
| IMPORT_DATE | Date of import EDBS File |

Tabelle 2-5 EDBS_DATAFILES

2.5 Verarbeitung bundeslandspezifischer Objektkataloge

Es werden alle Folien und OS ausgewertet. Eine Einschränkung nur auf Objekte der ALK findet nicht statt. D.h. es können auch EDBS Daten importiert werden die sich nicht auf ALK Objektkataloge beziehen.

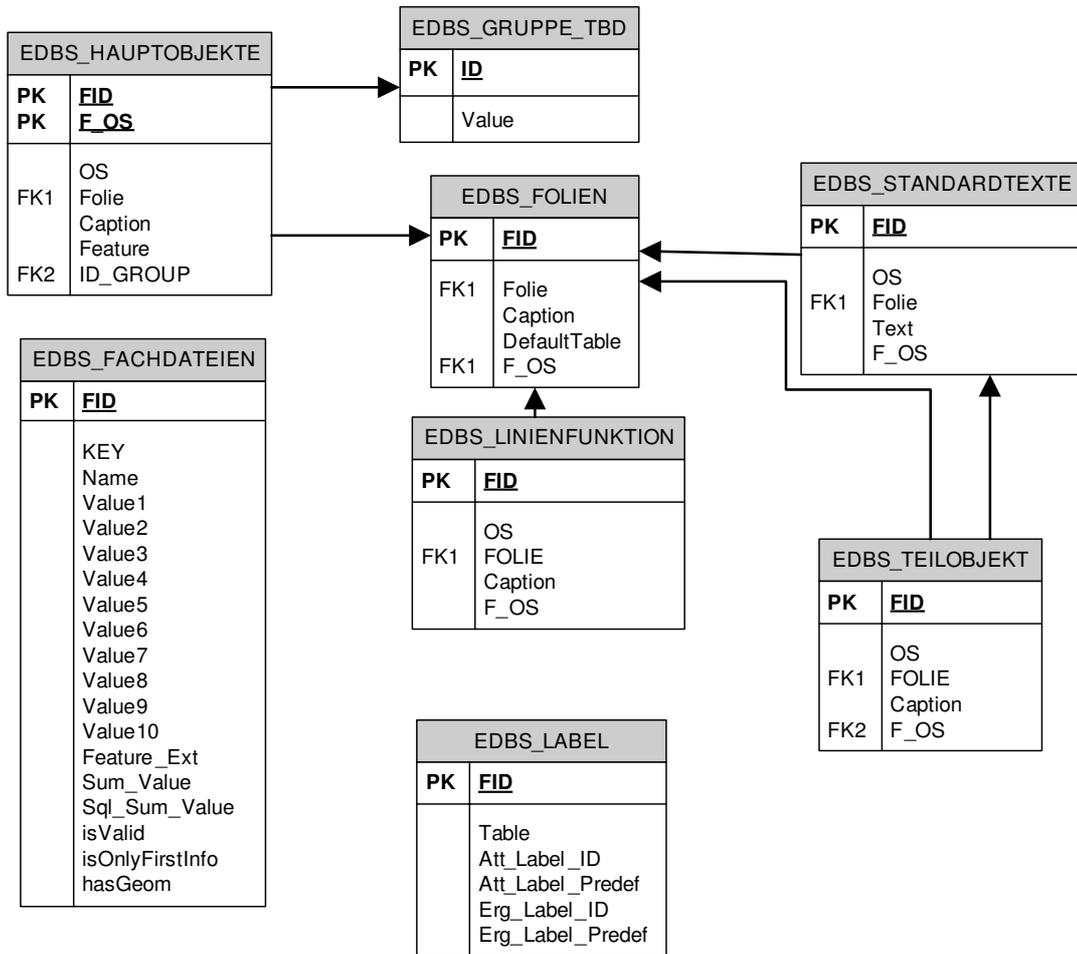
Der EDBS Import wird mit bundeslandspezifischen Kennungslisten und Zuordnungen geliefert. Fehlende OS, Folien und Fachdateien müssen durch den Benutzer in der Domain Control Table definiert werden. Objekte mit keiner Definition in der DCT werden nicht importiert.

Der Benutzer kann den Import unerwünschter Informationen –z.B. Schraffuren- ausschliessen.

Der EDBS Import verfügt über einen Mechanismus unbekannte OS, Folien und Fachdateien aus der EDBS Datei zu ermitteln und aufzulisten.

Topobase EDBS: Specification

2.5.1 Domain Control Table_[MRA5]



Eine Besonderheit ist das Feld F_OS. Hierbei handelt es sich um einen zusammengesetzten Schlüssel aus Folie und OS. Er hat das Format 001_0233. Hiermit wird eine eindeutige Abbildung möglich, da in einigen Bundesländern gleiche OS auf verschiedenen Folien unterschiedliche Bedeutung haben. Der Wert F_OS findet sich auch als Elementarer Schlüssel in jeder Feature Tabelle wieder. Hiermit wird die Fachbedeutung gesteuert, Listenfelder gesetzt und die Darstellung geregelt. Dadurch können auch Elemente importiert werden, für die die exakte Bedeutung zum Zeitpunkt des Imports nicht bekannt ist.

EDBS Hauptobjekte

| Column | Description |
|----------|---|
| FID | Primary Key |
| F_OS | Kombischlüssel Folie + OS (z.B. 001_0233) |
| Folie | ALK Folie |
| Caption | Beschreibung |
| Feature | Topobase ALK FC |
| ID_Group | ALK Gruppe - Topic |

Tabelle 2-6

EDBS Gruppe TBD

| <i>Column</i> | <i>Description</i> |
|---------------|--------------------|
| ID | Primary Key |
| Value | Wert |

Tabelle 2-7

EDBS Folien

| <i>Column</i> | <i>Description</i> |
|---------------|---|
| FID | Primary Key |
| Folie | ALK Folie |
| Caption | Beschreibung |
| DefaultTable | Wenn kein Eintrag für Feature dann hier die Zieltabelle |
| F_OS | Kombischlüssel Folie + OS (z.B. 001_0233) |

Tabelle 2-8

EDBS STANDARDTEXTE

| <i>Column</i> | <i>Description</i> |
|---------------|---|
| FID | Primary Key |
| OS | ALK Objektschlüssel |
| FOLIE | ALK Folie |
| F_OS | Kombischlüssel Folie + OS (z.B. 001_0233) |
| Text | ALK Text |

Tabelle 2-9

EDBS LINIENFUNKTION

| <i>Column</i> | <i>Description</i> |
|---------------|---|
| FID | Primary Key |
| OS | ALK Objektschlüssel |
| FOLIE | ALK Folie |
| CAPTION | Beschreibung |
| F_OS | Kombischlüssel Folie + OS (z.B. 001_0233) |

Tabelle 2-10

EDBS TEILOBJEKT

Topobase EDBS: Specification

| <i>Column</i> | <i>Description</i> |
|---------------|---|
| FID | Primary Key |
| OS | ALK Objektschlüssel |
| FOLIE | ALK Folie |
| Caption | Beschreibung |
| F_OS | Kombischlüssel Folie + OS (z.B. 001_0233) |

Tabelle 2-11

EDBS LABEL

| <i>Column</i> | <i>Description</i> |
|----------------|---------------------------------------|
| FID | Primary Key |
| Table | Topobase ALK Featureklasse |
| ATT_LABEL_ID | Label ID für Attributtabelle (_ATT) |
| ATT_LABEL_Pref | Labeldefinition |
| Erg_Label_ID | Label ID für Ergänzungstabelle (_ERG) |
| Erg_Label_Pref | Labeldefinition |

Tabelle 2-12

EDBS Fachdateien

| <i>Column</i> | <i>Description</i> |
|-----------------|--|
| KEY | Art der Information |
| NAME | Besondere Objektinformation |
| Value1-Value10 | Zielfeld für die jeweilig Objektinformation |
| Feature_Ext | Zieltabelle (z.B. _ATT) Info wird in Attributtabelle gespeichert |
| Sum_Value | Zielfeld für zusammengesetzte Information |
| SQL_Sum_Value | Zusammengesetzte Information aus Value 1-10) |
| isValid | |
| isOnlyFirstInfo | Label nur bei erster Objektinformation |
| hasGeom | Objektinformation besitzt Geometrie (z.B. Zuordnungspfeil) |

Tabelle 2-13

Die Folien der ALK sind im OSKA LiegKAT vergeben. Sie sind dreistellig und in der Zukunft eventuell auch mit einem alphanumerischen Zeichen.