SARACUS CONSULTING

DWx2r XSD/XML-Processing

Testversion

INHALTSVERZEICHNIS

1. DWx2r XSD/XML-Processing

DWx2r XSD/XML-Processing ist eine Utility, die generisch:

- aus einem XSD-Schema ein relationales Schema erzeugt,
- und XML-Dateien, die diesem Schema genügen, in eine relationale Datenbank lädt.

Bei der Schemaerzeugung gelten folgende Grundsätze:

- 0-1 Beziehungen (maxOccurs=1) zwischen XSD Parent-/Child-Strukturen (complexType) werden in die (relationale) Parent-Tabelle ‚gehoben'
- 0-n Beziehungen werden ausnormalisiert
- Mehrfach genutzte Unterstrukturen werden in einer gemeinsamen Tabelle abgelegt.

Ein einzelnes Top-Level-Element in einer XML-Datei wird im Folgenden auch als ‚Message' bezeichnet.

Die erzeugten Tabellen enthalten die folgenden technischen Attribute:

- dwh_msgid eindeutige ID, die eine XML-Message kennzeichnet
- dwh_id eindeutige ID, die jeden Record einer Message kennzeichnet
- dwh_pid die Parent-Id bei ausnormalisierten Unterstrukturen
- dwh_pathid der komplette XML-Pfad zu einem Element

Die Test-Version hat die folgenden Restriktionen:

- Lesen der Messages aus XML-Dateien, das Lesen der Messages aus Quelltabellen ist nicht unterstützt.
- Kein Multithreading, es gibt im Gegensatz zur Vollversion nur einen ‚Schreib'-Thread.
- Bei Persistieren von Messages legt der Schreib-Thread eine Denkpause von 500ms zwischen zwei Messages ein.
- Die Zieldatenbank muss eine ORACLE -Datenbank sein.

2. Installation und Konfiguration

Installation

- Es ist keine formale Installation notwendig.
- Die zip-Datei ist auf einem Rechner zu entpacken.
- Sie benötigen Java >= 1.7.
- Optional: eine Zieldatenbank die per JDBC erreicht werden kann.
    - Steht keine Datenbank zur Verfügung, so kann nur die Tabellenstruktur der XSD erstellt und in dem Ordner „ddl“ als sql-Datei abgelegt werden.

Konfiguration

Alle Konfigurationen müssen von Ihnen in die Datei „DWx2r.properties“ geschrieben werden. Die Konfiguration der Test-Version ist auf ein Minimum beschränkt. Sie müssen die Datenbank-Verbindung und den Aufenthaltsort der XSD-Datei konfigurieren.

Datenbank-Verbindung Beispiel:

```
db.work.database    = dbName
db.work.user        = user
db.work.password    = password
db.work.schema      = schema
```

Hier sind Name der Datenbank, Anmelde-User und dessen Password sowie das Zielschema einzutragen.

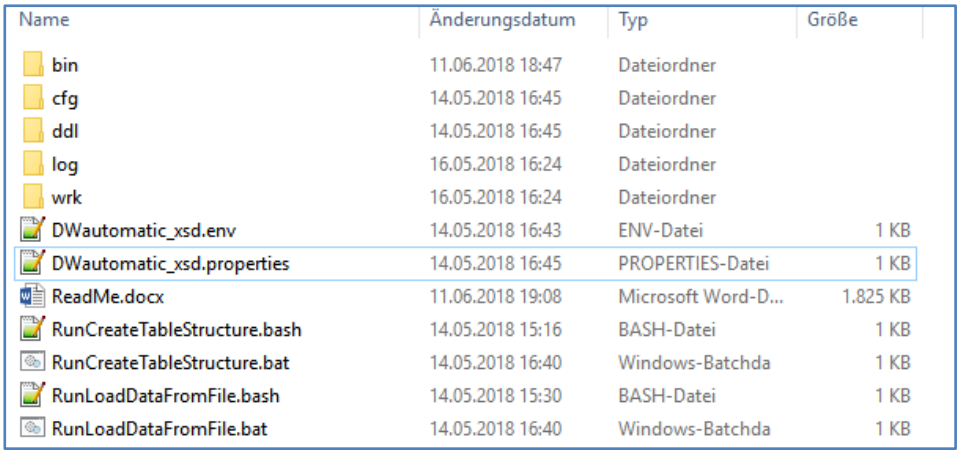
XSD-Datei Beispiel:

```
xsd.file            = Dateiname.xsd
xsd.element         = TopLevelElement
xml.string.db.size  = 2000
xml.string.db.type  = varchar2
xml.charset         = UTF-8
```

- Property `xsd.file` gibt den Dateipfad. Befindet sich die xsd-Datei **nicht** im root-Ordner, dann muss der komplette Pfad zur Datei gegeben werden.
- Property `xsd.element` gibt den Namen des *xsd:element* aus der xsd-Datei an.
- Property `xml.string.db.size` gibt die maximale Zeichenanzahl für String-Werte an. Die Größe wird genutzt, wenn keine Restriktion auf einem String-Element liegt.
- Property `xml.string.db.type` definiert den Datenbank-Typen, auf den Strings abgebildet werden.
- Property `xml.charset` gibt das Character-Set der zu lesenden XML-Datei an.

3. Aufruf

Der Test-Version liegen elementare Run-Scripts für Windows und Linux bei.



Name	Änderungsdatum	Typ	Größe
bin	11.06.2018 18:47	Dateiordner	
cfg	14.05.2018 16:45	Dateiordner	
ddl	14.05.2018 16:45	Dateiordner	
log	16.05.2018 16:24	Dateiordner	
wrk	16.05.2018 16:24	Dateiordner	
DWautomatic_xsd.env	14.05.2018 16:43	ENV-Datei	1 KB
DWautomatic_xsd.properties	14.05.2018 16:45	PROPERTIES-Datei	1 KB
ReadMe.docx	11.06.2018 19:08	Microsoft Word-D...	1.825 KB
RunCreateTableStructure.bash	14.05.2018 15:16	BASH-Datei	1 KB
RunCreateTableStructure.bat	14.05.2018 16:40	Windows-Batchda	1 KB
RunLoadDataFromFile.bash	14.05.2018 15:30	BASH-Datei	1 KB
RunLoadDataFromFile.bat	14.05.2018 16:40	Windows-Batchda	1 KB

Für Windows:

- RunCreateTableStructure.bat Erzeugen der Tabellenstruktur aus XSD
- RunLoadDataFromFile.bat Laden von XML-Messages aus Datei

Ausführung:

1. *PowerShell* oder *cmd* aufrufen.
2. Zum entpackten Ordner navigieren.
3. *RunCreateTableStructure.bat* per Eingabe über *PowerShell* oder *cmd* ausführen.

Für Linux:

- RunCreateTableStructure.bash Erzeugen der Tabellenstruktur aus XSD
- RunLoadDataFromFile.bash Laden von XML-Messages aus Datei
- DWx2r.env Gemeinsame Umgebungsvariablen

Ausführung:

1. *Terminal* aufrufen.
2. Zum entpackten Ordner navigieren.
3. *RunCreateTableStructure.*bash per Eingabe über das *Terminal* ausführen.

Falls die Standard-Javaversion kleiner als 1.7 ist, müssen die Scripts um den Pfad auf auf eine Javaversion >= 1.7 angepasst werden.

Für die Verzeichnisstruktur gilt:

- bin benötigte JAR-Dateien
- cfg sekundäre Konfigurationsdateien, werden automatisch erzeugt
- ddl erzeugte DDL wird hier abgelegt
- log Verzeichnis für Log-Dateien
- wrk Arbeitsverzeichnis
- smpl Beispiel xml- und xsd-Datei

3.1 Tabellen-Generierung

Die RunCreateTableStructure-Scripts erzeugen eine DDL-Datei, die manuell in die Datenbank geladen werden muss. Auf eine automatisierte Anlage der Tabellen wurde bewusst verzichtet.

Die DDL-Datei enthält neben den eigentlichen Zieltabellen auch die Dictionary-Tabellen. Diese Tabellen zeigen, welche Elemente/Attribute in welche Zieltabellen abgebildet wurden.

3.2 Laden der XML-Messages

Die RunLoadDataFromFile-Scripts laden Messages aus einer XML-Datei. Beim Aufruf eines Scripts ist hier der Pfadname der XML-Datei anzugeben.

3.3 Beispiel

In dem Ordner „smpl“ befindet sich ein Beispiel XML-Schema: shiporder.xsd

Mit folgenden Anpassungen in der Konfigurationsdatei „DWx2r.properties“ ist die Generierung einer beispielhaften Tabellenstruktur möglich.

```
xsd.file            = ./smpl/shiporder.xsd
xsd.element         = shiporder
```

Nach Anpassung der Konfigrationsdatei *RunCreateTableStructure.*bash bzw. *RunCreateTableStructure.*bat ausführen.