

MM Software Consulting GmbH



Kobalt • C ERP-DataServer

Dokumentation KbltSQL

Inhalt

1. Aufgabe des Programms	3
2. Installation	3
3. Kommandozeilenparameter.....	3
4. Aufbau und Syntax von *.ksql Dateien (DDL / Tabellenstruktur).....	4
5. Aufbau und Syntax von *.kdat Dateien (Daten / Tabelleninhalte).....	5
6. Konfigurationseinträge in der admin.conf	6
7. Logging	6
8. Release Notes	6
ab v3.17.09.6174	6

1. Aufgabe des Programms

Das Hilfsprogramm KbltSQL.exe dient im Kobalt-C Softwarepaket zur Wartung der Kobalt-Datenbank. Bei alle Datenmodell-Änderungen oder Versionsupdates wird dies Programm zur Datenbankpflege eingesetzt.

KbltSQL verarbeitet Kommandozeilen-Parameter, *.ksql und *.kdat Dateien aus dem Resources-Ordner der Kobalt-Installation.

2. Installation

Das Programm KbltSQL.exe wird mit dem Kobalt-C Installationsprogramm ausgeliefert und gehört zum Standardumfang des Softwarepakets. Eine gesonderte Installation oder Einrichtung ist nicht notwendig.

3. Kommandozeilenparameter

Wie bei allen Kobalt-C Programmen kann die Aufrufsyntax mit der Option --help abgerufen werden.

Option	Beschreibung
-k	Verarbeiten einer *.ksql-Datei. Wird kein kompletter Dateipfad angegeben, so sucht das Programm die Datei im Ordner <Kobalt-Inst>\Resources Beispiel: KbltSQL.exe -k test.ksql
--exec	wie die Option -k
-i --import	Verarbeiten einer *.kdat-Datei. Wird kein kompletter Dateipfad angegeben, so sucht das Programm die Datei im Ordner <Kobalt-Inst>\resources\data Beispiel: KbltSQL.exe -i daten.kdat
--dumtable	Ein Tabelleninhalt oder Teile davon werden in eine *.kdat Datei exportiert.
--dumpstmt	Legt die zu exportierenden Daten mittels --dumtable Option fest.
--outfile	Bestimmt den Dateinamen für die --dumtable Option. Wird kein explizierter Dateiname angegeben, so exportiert --dumtable in die Datei <Kobalt-Inst>\resources\data\dump_<schema>_<table>.kdat

--append	<p>Mit dieser Option kann festgelegt werden, das Exporte an eine ggf. schon vorhandene Datei angehängt werden.</p> <p>Beispiel: KbItSQL –dumtable kb.user</p> <p>Der Tabelleninhalt aus der Tabelle ‚kb.user‘ wird in die Datei <Kobalt-Inst>\resources\data\dump_kb_user.kdat exportiert.</p>
--ini	<p>Mit dieser Option kann eine von admin.conf abweichende Parameterdatei festgelegt werden. Die Datei M*U*S*S ebenfalls im <Kobalt-Inst>\conf Ordner stehen.</p>
-n --no_defaults	<p>Standardmäßig werden bei jedem KbItSQL-Programmlauf die Spach-Textbausteine und Modul-KSQL Dateien verarbeitet.</p> <p>Mit dieser Option wird nur noch das angegebene Kommando ausgeführt.</p>

4. Aufbau und Syntax von *.ksql Dateien (DDL / Tabellenstruktur)

Der Aufbau dieser Kommandodatei ist wie folgt:

Zeilen die mit #, // oder auch – beginnen werden als Kommentare angesehen und ignoriert.

Dies ist ein Kommentar

```
ksql {
  DB-Context: <frei definierbarer Kontextname>
  File-Version: <Nummer>
}
```

```
ksql {
  ver: <Nummer|direct>
  eng:pg
  stmt:
  SQL Anweisung
}
```

... weitere ksql { } Blöcke

Nach jeder Verarbeitung wird unter dem Schlüssel ‚DB-Context‘ die höchste registrierte ver:-Nummer abgespeichert. Wird eine weitere oder auch dieselbe Datei mit gleichem ‚DB-Context‘ ausgeführt, so werden nur ksql{}-Blöcke verarbeitet, bei denen die ver:-Nummer größer als die bereits registrierte ist. Bei ksql{}-Blöcken mit einer ‚ver: direct‘-Klausel, wird der Block sofort nach dem einlesen

verarbeitet und eine Versionsüberprüfung übergangen.

Die DB-Context Daten werden in den Tabellen kb.version und kb.versionstmt abgelegt.

Ab der Version 3.17.09.6174 besteht die Möglichkeit, einen Tabelleninhalt in eine kdat-Datei zu exportieren. Will man z.B. alle Datensätze aus der Tabelle kb.user in eine Datei 'c:\temp\user.kdat' exportieren, so wäre folgender ksql{}-Block zu formulieren:

```
ksql {  
  ver: <Nummer|direct>  
  eng:pg  
  dumptable: kb.user  
  outfile: c:\temp\user.dat  
  stmt:  
  select * from kb.user  
}
```

5. Aufbau und Syntax von *.kdat Dateien (Daten / Tabelleninhalte)

Der Aufbau dieser Datendateien ist wie folgt:

Zeilen die mit #, // oder auch – beginnen werden als Kommentare angesehen und ignoriert.

Einzeilige SQL-Anweisungen wie z.B. delete from <tabelle> where <xyz>

stmt:

SQL-Anweisung

execute

mehrzeilige SQL-Anweisungen wie z.B.

insert into <tabelle> (feld1, feld2)

values ('x', 'y')

stmt:

SQL-Anweisung

mit weitere Zeile...

execute

Bulk-Importe

stmt:

insert into <table> (f1,f2,f3) values (\$1,\$2,\$3)

data-rows:

"a1";"a2";"a3 "

"b1";"b2";"b3 "

"c1";"c2";"c3 "

...usw.

6. Konfigurationseinträge in der admin.conf

Das Programm verwendet die Einstellparameter aus der admin.conf Datei. Es werden nur die Standardparameter aus den Segmenten [db], [Logger] und [log] genutzt.

(Die admin.conf Parameter werden in der Dokumentation Kobalt_C_admin_conf.pdf aufgeführt.)

7. Logging

Das Programm schreibt Meldungen in eine Tagesdatei in den Ordner 'log' der Kobalt-C Installation mit dem Dateinamen KbItSQL_<Datum>.log .

Ältere Log-Dateien werden in den Archivordner des Installationspfades verschoben.

8. Release Notes

ab v3.17.09.6174

Tabelleninhalte (dumps) können sowohl in *.ksql-Dateien als auch über die Kommandozeile erzeugt werden.

Befehle dazu:

dumptable:	<Tabellenname>	welche Tabelle sichern
outfile:	<Dateiname/Pfad>	in welche Datei sichern
append:	true false	Datei überschreiben(=false) oder anhängen(=true)
dumpstmt:	SQL-Ausdruck	was soll gesichert werden, z.B. ‚select * from tabelle‘ für die ganze Tabelle