

IBM DB2 Web Query 2.2.1 – WQRepTab - Tool zur Ermittlung von Reports mit Tabellen Teil-3

Da Db2 Web Query bisher kein Feature bietet, anhand dessen es möglich ist, eine Liste von Reports mit den darin verwendeten Dateien zu erstellen, habe ich begonnen, selbst ein Tool mit dem Namen WQRepTab zu erstellen. Das Tool selbst ist noch nicht perfekt, bietet aber einen ersten Einstieg. Geplant ist eine Weiterentwicklung des Tools, an dessen Prozess ich Sie gerne teilhaben lassen möchte.

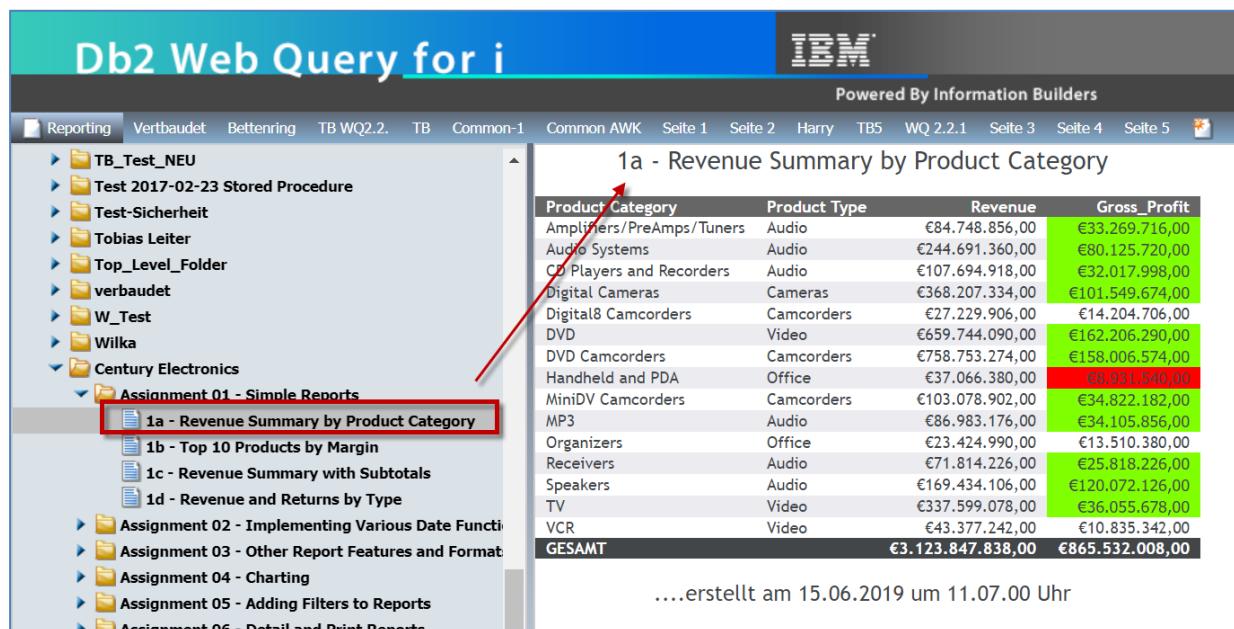
In den letzten Artikeln habe ich Ihnen eine Übersicht über den Ablauf des Tools WQRepTab gegeben. Im aktuellen Artikel wird die technische Erläuterung des Tools fortgesetzt und beendet.

Das Tool WQRepTab

Das Tool WQRepTab ermittelt zunächst die vorhandenen Reports. Unter Reports sind alle Arten von Auswertungen zu verstehen, die mit Web Query erstellt wurden: Berichte, Diagramme, Dokumente, Dashboards und Visualisierungen.

Über mehrere Schritte werden dann der Pfad für das verwendete Synonym und daraus dann die verwendeten Dateien ermittelt.

Zurzeit werden nur Reports korrekt analysiert, die ein Synonym und maximal fünf Dateien bzw. Tabellen verwenden. Siehe dazu auch das Kapitel *Was ist noch offen* und *Geplante Erweiterungen*.



The screenshot shows the Db2 Web Query interface. The left sidebar contains a navigation tree with various folders and reports. A red box highlights the report '1a - Revenue Summary by Product Category'. The main content area displays the report's title and a table with data. A red arrow points from the highlighted report in the sidebar to the table in the main area. The table has columns: Product Category, Product Type, Revenue, and Gross Profit. The data includes items like Amplifiers/PreAmps/Tuners, Audio Systems, CD Players and Recorders, Digital Cameras, etc. The table ends with a 'GESAMT' row. Below the table, a timestamp is visible: '....erstellt am 15.06.2019 um 11.07.00 Uhr'.

Product Category	Product Type	Revenue	Gross Profit
Amplifiers/PreAmps/Tuners	Audio	€84.748.856,00	€33.269.716,00
Audio Systems	Audio	€244.691.360,00	€80.125.720,00
CD Players and Recorders	Audio	€107.694.918,00	€32.017.998,00
Digital Cameras	Cameras	€368.207.334,00	€101.549.674,00
Digital8 Camcorders	Camcorders	€27.229.906,00	€14.204.706,00
DVD	Video	€659.744.090,00	€162.206.290,00
DVD Camcorders	Camcorders	€758.753.274,00	€158.006.574,00
Handheld and PDA	Office	€37.066.380,00	€8.931.540,00
MiniDV Camcorders	Camcorders	€103.078.902,00	€34.822.182,00
MP3	Audio	€86.983.176,00	€34.105.856,00
Organizers	Office	€23.424.990,00	€13.510.380,00
Receivers	Audio	€71.814.226,00	€25.818.226,00
Speakers	Audio	€169.434.106,00	€120.072.126,00
TV	Video	€337.599.078,00	€36.055.678,00
VCR	Video	€43.377.242,00	€10.835.342,00
GESAMT		€3.123.847.838,00	€865.532.008,00

Abbildung 1: Mit Db2 Web Query erstellter Report

Betrachten wir den in Abbildung 1 dargestellten Report mit dem Titel „1a – Revenue Summary by Product Category“.

Sie wollen nun wissen, welche Dateien oder Tabellen für diesen Report verwendet werden. Diese Information liefert Ihnen das Tool WQRepTab.

WQRepTab

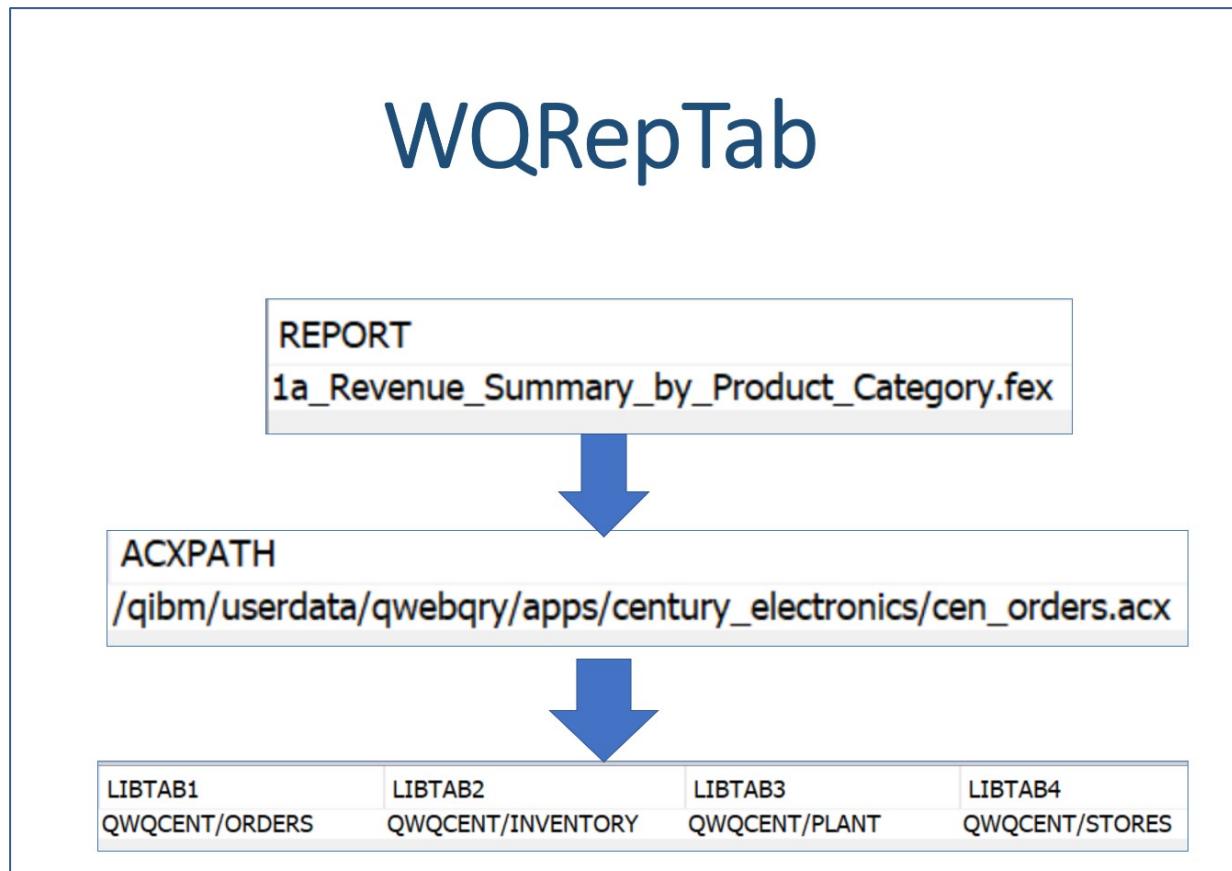


Abbildung 2: Wie das Tool WQRepTab die Tabellen eines Reports ermittelt

Programmablauf

Das Basisprogramm von WQRepTab ist das ILE-Programm WQ00. Darin werden einige Vorbereitungen für das Tool getroffen und der komplette Ablauf gesteuert.

Die Teile (A) bis (L) wurden in den letzten Artikeln bereits beschrieben. Ich setzte die Beschreibung mit dem Teil (M) in Abbildung 7 fort.

Programm WQ00

```
**Free                                         (A)
// ****
// ** Programm: WQ00
// ** Autor:   Theo Bär, EDV-Beratung
// **           Ringmauerweg 1
// **           69250 Schönaeu / Heidelberg
// **           E-Mail: Theo.Baer@edv-baer.com
// **           Tel.: 06228 / 912 630
// **           ***
// ** Zweck:   Hauptprogramm für das Tool WQRepTab
// **           ***
// ** erstellt: 06.2019
// **           ***
// ** geändert:
// **           ***
// ****

ctl-opt Option(*NoDebugIO: *SrcStmt)          (B)
actgrp(*new)
AlwNull(*UsrCtl) ;

dcl-pr wq01 extpgm('WQREPTAB/WQ01') End-Pr;   (C)
dcl-pr wq02 extpgm('WQREPTAB/WQ02') End-Pr;
dcl-pr wq03 extpgm('WQREPTAB/WQ03') End-Pr;

dcl-pr Run      ExtPgm('QCMDEXC');
CmdStr  Char(3000) Options(*VarSize);
CmdLen  Packed(15:5) Const;
CmdDbcs Char(2)      Const Options(*Nopass);
End-pr;

dcl-s CmdStr      char(3000);
```

Abbildung 3: Definitionen für das Basisprogramm WQ00 – Teil 1

Programm WQ03

```
// Tabellen pro Report ermitteln (M)
wq03();

dsply 'Tables for every report created';

// 
*InLr = *On;
Return;

//
```

Abbildung 4: Programm WQ03 aufrufen und Programm WQ00 beenden

(M) Das Programm WQ03 wird aus WQ00 aufgerufen

```
// Namen der verwendeten Tabellen aus Spalte WQSYNDAT ermitteln
// 

dcl-f wqsyn  usage(*update) (M1)
          extdesc('WQREPTAB/WQREPSYN')
          extfile(*extdesc)
          rename(wqrepsyn: wqsynr);

dcl-s lib1 char(10); (M2)
dcl-s tab1 char(10);
//dcl-s libTab1 char(21);
dcl-s anzrep zoned(5:0); // Anzahl Reports

dcl-ds wqsynDS extname('WQREPTAB/WQREPSYN'); (M3)
  libTab char(21) dim(5) pos(1701);
// libTab char(21) dim(5) samepos(LibTab1); // nur mit IBMi 7.3 TR6
End-Ds;

  read wqsyn; (M4)
  dow not %eof(wqsyn);
    anzrep += 1;
    dsply ('Report Nummer: ' + %char(anzrep));
    suchsr();
    read wqsyn;
  EndDo;

*InLr = *on;
Return;

//-----// (M5)

dcl-proc suchsr;
  dcl-s pos1 zoned(5:0) inz(1);
  dcl-s pos2 like(pos1) inz(1);
  dcl-s anz zoned(2: 0)inz(0); // Anzahl Tabellen pro Report

  dsply %subst(Report: 1: 50);

// Elemente suchen, z.B. TABLENAME=PROFILDT/ARTSP,
  dow pos1 > 0;
    pos1= %scan('TABLENAME=': WQSYNDAT: pos2); //
```

```

if pos1 = 0;
  if anz > 0;
    dsply 'Keine weitere Tabelle vorhanden';
    update wqsynr;

    leave;
  EndIf;
  if anz = 0;
    dsply 'Keine Tabelle vorhanden';
    anz = 0;
    leave;
  EndIf;
endif;

dsply pos1;

pos2 = %scan(',', : WQSYNDAT: pos1 + 1);
dsply pos2;

anz += 1;
dsply ('Anzahl Tabellen für Report: ' + %char(anz));
libTab(anz) =
  %trim(%subst(wqsyndat: pos1+10 : pos2 - (pos1+10)));
dsply libTab(anz);

enddo;

End-Proc;

```

Abbildung 5: Das Programm WQ03

(M1) Die Datei WQSYN (= WQREPSYN) wird als Update-Datei definiert.

(M2) Definition verschiedener Variablen.

(M3) Die extern beschriebene Datenstruktur verweist auf die Datei WQSYNDAT. Für die folgende Verarbeitung werden die fünf Felder LIBTAB1 bis LIBTAB5 durch ein Array LIBTAB mit Dimension 5 überlagert. Zunächst wird dies mit dem Schlüsselwort POS(1701) realisiert.

Ab IBM i 7.3 Technology Refresh 6 kann das Schlüsselwort SAMEPOS(LIBTAB1) mit dem ersten zu überlagernden Feld verwendet werden. Falls im Nachhinein weitere Spalten vor LIBTAB eingefügt werden, ist dann keine Änderung der zu überlagernden Position erforderlich.

(M4) Es werden alle Sätze der Datei WQSYN (= WQSYNDAT) verarbeitet. Für jeden Satz wird die Prozedur SuchSR() aufgerufen.

(M5) Die Prozedur sucht alle Einträge mit der Zeichenfolge „TABLENAME=“. Die anschließende Zeichenfolge liefert den Namen der Datei. Der Name endet mit einem Komma.

Ende des Programms WQ00

```

// Report-Titel + Dateien auflisten
EXEC SQL
  Create table qtemp.wqrep as (
    select report, libtab1, libtab2, libtab3, libtab4, libtab5 from
  wqreptab.wqrepsyn)
    with data;

CmdStr = 'RUNQRY () qtemp/wqrep';

```

(N)

```
Run(Cmdstr:%Size(CmdStr));  
  
*InLr  = *On;  
  
Return;
```

Abbildung 6: Ende des Programms WQ00

(N) Am Ende des Haupt-Programms WQ00 wird noch eine temporäre Datei in QTEMP mit den zuvor ermittelten Daten in WQREPSYN erstellt. Diese Datei wird dann mit RUNQRY angezeigt.

Ausführung von WQRepTab

Zur Ausführung von WQRepTab fügen Sie die Bibliothek WQREPTAB der Bibliotheksliste hinzu:
ADDLIBLE WQREPTAB

Starten Sie dann das zentrale Programm WQ00 mit:
CALL WQ00

Für die einzelnen Programmschritte wird mit DSPLY jeweils eine Information angezeigt. Falls das Programm innerhalb von PDM ausgeführt wird, bleibt das Programm bei jeder DSPLY-Anweisung stehen und wird erst dann fortgesetzt, wenn die Enter-Taste gedrückt wird.

Bei Ausführung des Programms außerhalb von PDM läuft das Programm auch nach einer DSPLY-Anweisung weiter, ohne dass die Enter-Taste gedrückt werden muß. Ausnahme davon ist, sobald eine Query-Abfrage innerhalb des Programms ausgeführt wird.

Und so sieht dann bei mir das Ergebnis aus:

REPORT	LIBTAB1	LIBTAB2	LIBTAB3	LIBTAB4	LIBTAB5
06b_Erlöse20172016Differenz.fex	... REVELL/REPVKP0				
07a_Pivot.fex	... REVELL/REPVKP0				
07a_Pivot_OK.fex	... REVELL/REPVKP0				
8a_Diagramm_Erlöse_Kundengruppe.fex	... REVELL/REPVKP0				
8b_Kreis_Diagramm_Erlöse_Kundengruppe.fex	... REVELL/REPVKP0				
9a_Dokument.fex	... REVELL/REPVKP0				
9b_Dashboard.fex	... REVELL/REPVKP0				
3a_Report_mit_Across_für_das_Jahr_12_Jahre.fex	... QWQCEN/INVENTOR...				
8a_Report_mit_variablen_Jahren_12_Jahre..fex	... QWQCEN/INVENTOR...				
Report_mit_Gruppierung_ProductCategory.fex	... QWQCEN/ORDERS ... QWQCEN/IN... QWQCEN/PL... QWQCEN/ST...				
Report10Zwischensummen_und_.fex	... QWQCEN/ORDERS ... QWQCEN/IN... QWQCEN/PL... QWQCEN/ST...				
Report1Umsatz.fex	... QWQCEN/ORDERS ... QWQCEN/IN... QWQCEN/PL... QWQCEN/ST...				
Heute_Erster_Vormonat.fex	... QWQCEN/ORDERS ... QWQCEN/IN... QWQCEN/PL... QWQCEN/ST...				
Erster_Vormonat_Synonym.fex	... QWQCEN/ORDERS ... QWQCEN/IN... QWQCEN/PL... QWQCEN/ST...				
1_Report_Akkordeon_Across_Jahr.fex	... QWQCEN/ORDERS ... QWQCEN/IN... QWQCEN/PL... QWQCEN/ST...				
Report1.fex	... QWQCEN/ORDERS ... QWQCEN/IN... QWQCEN/PL... QWQCEN/ST...				
Report2.fex	... QWQCEN/ORDERS ... QWQCEN/IN... QWQCEN/PL... QWQCEN/ST...				
Report3.fex	... QWQCEN/ORDERS ... QWQCEN/IN... QWQCEN/PL... QWQCEN/ST...				
Report4.fex	... QWQCEN/ORDERS ... QWQCEN/IN... QWQCEN/PL... QWQCEN/ST...				
Report5.fex	... QWQCEN/ORDERS ... QWQCEN/IN... QWQCEN/PL... QWQCEN/ST...				
Report6.fex	... QWQCEN/ORDERS ... QWQCEN/IN... QWQCEN/PL... QWQCEN/ST...				
Report7.fex	... QWQCEN/ORDERS ... QWQCEN/IN... QWQCEN/PL... QWQCEN/ST...				
Report8.fex	... QWQCEN/ORDERS ... QWQCEN/IN... QWQCEN/PL... QWQCEN/ST...				
Lightweight1.fex	... QWQCEN/ORDERS ... QWQCEN/IN... QWQCEN/PL... QWQCEN/ST...				
Lightweight2_MultiDiagramm.fex	... QWQCEN/ORDERS ... QWQCEN/IN... QWQCEN/PL... QWQCEN/ST...				
Lightweight3_Welt.fex	... QWQCEN/ORDERS ... QWQCEN/IN... QWQCEN/PL... QWQCEN/ST...				

Abbildung 7: Liste der Reports mit den verwendeten Tabellen

Beschränkungen von WQRepTab

- Die Längen der einzelnen Felder von WQREPDAT sind zurzeit wie folgt definiert und damit auch längenmäßig begrenzt:
 - Report-Titel ist char(100)
 - Name des Synonyms ist char(200)
 - Adresse des IFS-Pfads für das Synonym ist char(400)
 - Das Feld für den Inhalt des ACX-Synonyms ist char(1000)
 - Die Felder für die Dateien sind char(21) für Bibliothek (10), Trennzeichen /(1) und Dateiname (10)Für meine Beispiele waren die Längenangaben bisher ausreichend.
- Pro Report können maximal fünf Dateien angegeben werden.

- Zurzeit werden nur die Synonyme bei Dateiverwendung erkannt und behandelt. Eine Analyse für Stored Procedures, SQL-Skripte, Services o.ä. fehlt noch.

Was ist noch offen?

Hinweis / Fehlermeldung, falls der erzeugte IFS-Pfad nicht existiert (siehe bei mir der Ordner „porter“. Wie kann das überhaupt passieren?)

Geplante Erweiterungen

Die aktuelle Version von WQRepTab berücksichtigt bei der Ermittlung der Synonyme nur das erste gefundene Synonym. Falls weitere Synonyme für einen Report verwendet werden, werden diese nicht berücksichtigt.

Zurzeit werden maximal fünf Tabellen pro Report gespeichert. Dies kann bei Bedarf erweitert werden.

Kennzeichnung der Berichte, falls keine Tabellen sondern andere Objekte verwendet werden wie Stored Procedures, SQL-Skripte, Services o.ä.

Erweiterung oder Anpassung der Feldlängen für Report-Titel, IFS-Pfad, Synonym-Text.

Zurzeit sind die Felder der Tabelle WQSYNDAT als Char-Felder definiert. Die Definition wird demnächst in Varchar geändert. Dabei ist zu beachten, dass die Definition der Feldgruppe LibTab in WQ03 angepasst werden muss:

```
dcl-ds wqsynDS extname('WQREPTAB/WQREPSYN');
  libTab char(21) dim(5) pos(1701);
End-Ds;
```

Könnte bei POS auch der Name des ersten zu überlagernden Feldes verwendet werden? Laut Beschreibung im Referenzhandbuch ist das leider nicht möglich:

The POS keyword is used in a free-form subfield definition to specify the starting position of the subfield in the data structure.

The starting-position parameter must be a value between 1 and the length of the data structure.

The parameter can be a literal or a named constant. If it is a named constant, the constant must be defined prior to the definition statement.

Aber inzwischen gibt es dafür quach eine Lösung:

Mit Technology Refresh TR6 IBM i 7.3 wurde ein neues Schlüsselwort eingefügt: SAMEPOS. Als Parameter für SAMEPOS wird der Name eines Feldes angegeben, das zuvor in der Datenstruktur definiert wurde. Unsere Datenstruktur wqsynDS würde dann folgendermaßen aussehen:

```
dcl-ds wqsynDS extname('WQREPTAB/WQREPSYN');
  libTab char(21) dim(5) samepos(LibTab1);
End-Ds;
```

Die Auswertungen werden zurzeit nur mit Query/400 erstellt. Zusätzlich ist eine Realisierung mit Web Query geplant.

Die Datei WQREPSYN soll noch um eine Spalte erweitert werden, die den Autor des Reports enthält. (siehe Datei qwqrepos.Wf_Reposob, Spalte CREATEDBY)

Auswertung nach Hauptordner, siehe:

select * from qwqrepos.wf_reposobj order by prt_path, objname; (enthält Struktur Repository

Haupt-Ordner, Unter-Ordner...Report-Name.

Anzeigen „sichtbaren“ Titel, nicht Fex-Namen (Datei qwqrepos.WF_NLSOBJ, Spalte OBJDESC).

Hinweis:

Falls Sie Interesse an diesem Tool haben, schicken Sie mir einfach eine E-Mail:

Theo.Baer@edv-baer.com

Vorschau:

Im nächsten Artikel zeige ich Ihnen verschiedene Auswertungen von WQRepTab, die mit Db2 Web Query erstellt wurden.

Bis dahin wünsche ich Ihnen weiterhin viel Spaß beim Vermehren Ihrer Fertigkeiten.

Den Autor Theo Bär erreichen Sie unter EDV-Beratung Theo Bär - Ringmauerweg 1 - 69250 Schönau - Tel. (+49) 6228 912 630 - e-Mail info@edv-baer.com