



WireLess Mobile XML-DBL V2

New XML-D (XML-DBL) For DBL file design

1	Previous system	3
2	New system.....	3
2.1	Enhancements	3
2.2	Usage.....	3
3	Environment.....	3
4	File Description.....	3
5	Elements.....	4
5.1	DBLFILE element tag.....	4
5.1.1	Description.....	4
5.1.2	Fields.....	4
5.1.3	Data.....	4
5.1.4	Usage.....	4
5.2	DATABASE element tag.....	4
5.2.1	Description.....	4
5.2.2	Fields.....	4
5.2.3	Data.....	4
5.2.4	Usage.....	5
5.3	SQL element tag	5
5.3.1	Description.....	5
5.3.2	Fields.....	5
5.3.3	Data.....	5
5.3.4	5
5.3.5	Usage.....	5
5.4	DATA-ADD element tag.....	5
5.4.1	Description.....	5
5.4.2	Fields.....	5
5.4.3	Data.....	6
5.4.4	Usage.....	6
5.5	DATA-UPDATE element tag.....	6

5.5.1	Description.....	6
5.5.2	Fields.....	6
5.5.3	Data.....	7
5.5.4	Usage.....	7
5.6	DATA-REPLACE element tag.....	7
5.6.1	Description.....	7
5.6.2	Fields.....	7
5.6.3	Data.....	8
5.6.4	Usage.....	8
5.7	DATA-DELETE element tag.....	8
5.7.1	Description.....	8
5.7.2	Fields.....	8
5.7.3	Data.....	8
5.7.4	Usage.....	8
6	Reserved keywords tags (further use).....	9
6.1	VERSION element tag.....	9
6.1.1	Description.....	9
6.1.2	Fields.....	9
6.1.3	Data.....	9
6.1.4	Usage.....	9
6.2	OPTIONS element tag.....	9
6.2.1	Description.....	9
6.2.2	Fields.....	9
6.2.3	Data.....	9
6.2.4	Usage.....	9
7	Samples.....	11

1 Previous system

A text file having a ".dbl" extension is used to send database data from server to client. The DBL file is a text file (tab separated) and all rows are inserted in the database.

The DBL file has 3 data columns (separated by tabs) as contents of database fields "REF", "NAM" and "Others".

The DBL file is built by the Wmb server (by the Inchecker module).

2 New system

The new DBL file is designed to send data in a more flexible system, and execute SQL queries. The version 2 add new functionalities (Data-replace) and new enhancements in existing tags (multiple key search).

2.1 Enhancements

SQL execution queries

Database file selection.

Table selection.

Field selection.

Add, replace, update or delete data records.

Key fields selection and data column selection for searches.

2.2 Usage

The new system is used in the same environment, but the DBL file is not built by the server yet.

The file is transferred from server to client during synchronization.

After processing the ".dbl", the XML-D input file is deleted.

3 Environment

The XML-DBL file would be generated by a script server-side and stored in the data channel folder.

The name of the file is *ChannelName.DBL*. *ChannelName* is the name of the current channel to synchronize.

4 File Description

"XML style element Tag" as data encapsulation.

One character to escape prefix tag (<).

One character to escape suffix tag (>).

CR/LF (0x0D 0x0A) breaks the data lines.

One character as data column separator (| or "pipe").

Tags begin with "tag prefix" (<). Command keyword (tag name) follows prefix.

Command keyword (tag) applies to data until reach closing /command keyword (closing tag).

Command prefix is escaped into data as <.

Field separator is escaped in data (\\)

As (not yet implemented) Option :

All control chars (0x00-0x1F) in data are uncontroffied (escape prefix followed by char+20h)

5 Elements

5.1 DBLFILE element tag

5.1.1 Description

The file element contains all other elements in the DBL file.

First / last element in file (consistency control).

5.1.2 Fields

The DBLFILE element has no fields.

5.1.3 Data

The DBLFILE element has no data.

5.1.4 Usage

```
<DBLFILE>
```

```
... all other tags ...
```

```
</DBLFILE>
```

5.2 DATABASE element tag

5.2.1 Description

Sets the SDF database used for the following updates (Only SQL-CE supported).

One or more DATABASE elements can exist in the dbl file.

5.2.2 Fields

PATH

The full path and file name of the database.

Optional, if the database name is not set, the database used is the standard channel database.

TYPE (obsolete)

The database engine type.

"SQLCE" for SQL server for Win CE.

5.2.3 Data

The DATABASE element has no data.

5.2.4 Usage

```

<DBLFILE>
<DATABASE PATH="\windows\wm\mychannel\mybase.sdf" TYPE="SQLCE">
...
One or more SQL, DATA-ADD, DATA-DELETE and DATA-UPDATE element.
...
</DATABASE>
<DATABASE>
...
One or more SQL, DATA-ADD, DATA-DELETE and DATA-UPDATE element.
...
</DATABASE>
<DBLFILE>

```

5.3 SQL element tag

5.3.1 Description

The data of this element is an SQL query. The query is performed in the current database.

5.3.2 Fields

The DBLFILE element has no fields.

5.3.3 Data

Contains the SQL query. The query do not needs to end with a semicolon sign ;.

5.3.4

5.3.5 Usage

```

<DBLFILE>
<DATABASE>
<SQL>
SQL command
</SQL>
<SQL>SQL command</SQL>
</DATABASE>
<DBLFILE>

```

5.4 DATA-ADD element tag

5.4.1 Description

Data to add (insert) into a table in the current database.
No search is performed. If a record exists with a field having unique key, data will not be added (integrity violation).

5.4.2 Fields

TABLE

Optional. The table name in current database.
If not supplied, the table name used is "ITEMS".

Fieldxx

Optional. Field identifier.

If field names are supplied, they must be the same quantity as columns in each row.
Each field identifier (FIELDxx) must be unique. The "xx" can have any value. The order of field identifiers is significant and is relative to the data supplied.

5.4.3 Data

The column data to fill the table fields, separated by "|" (pipe character).

The number of data columns should be the same of fields in the table.

If the field names are not supplied, the data is set into fields in the same order of the table definition.

5.4.4 Usage

<DBLFILE>

<DATABASE>

<DATA-ADD TABLE="ITEMS" FIELD01="REF" FIELD02="NAM">

001|NamOfMyRef1

002|NamOfMyRef2

</DATA-ADD>

<DATA-ADD TABLE="ITEMS">

001|MyNam1|Other1

002|MyNam2|Other2

</DATA-ADD>

<DATA-ADD >

001|MyNam1|Other1

002|MyNam2|Other2

</DATA-ADD>

</DATABASE>

<DBLFILE>

5.5 DATA-UPDATE element tag

5.5.1 Description

Data to update (amend) into existing records of a table in the current database.

The field(s) name(s) in the table denoted by FIELDnn are updated in the selected records of the table.

The field(s) name(s) in the table denoted by KEYFLDnn are used to select the record(s) in the table.

One or more fields must be supplied to update. One or more fields must be supplied to select the records. Database wildcards will be used in data columns.

5.5.2 Fields

TABLE

Optional. The table name in current database.

If not supplied, the table name used is "ITEMS".

FIELD OR FIELDnn

Mandatory. The field(s) name(s) in the table to update with new values.

Each field identifier (FIELDxx) must be unique. The "xx" can have any value. The order of field identifiers is significant and is relative to data supplied.

KEYFLD of KEYFLDnn

Mandatory. The field(s) name(s) in the database used to search the records. Each field identifier (KEYFLDxx) must be unique. The "xx" can have any value. The order of field identifiers is significant and is relative to data supplied.

5.5.3 Data

The data to update the table fields and search the records, separated by "|" (pipe character).

The number of data columns must be the same of the sum of FIELDxx and KEYFLDxx identifiers supplied.

5.5.4 Usage

```
<DBLFILE>
<DATABASE>
<DATA-UPDATE TABLE="ITEMS" FIELD00="NAM" FIELD01="OTHER"
KEYFLD="REF">
MyNam1|Other1|001
MyNam2|Other2|002
</DATA-UPDATE>
<DATA-UPDATE TABLE="CLIENTS" FIELD01="delai_liv" KEYFLD01="codcli">
02|008003
03|960273
</DATA-UPDATE>
</DATABASE>
<DBLFILE>
```

5.6 DATA-REPLACE element tag

5.6.1 Description

Data to replace (delete and add) into a table in the current database.

The field(s) name(s) in the table denoted by KEYFLDnn are used to search the record.

The data columns denoted by VALPOSnn (zero based) are used to search the record in the database.

All fields are used to fill the new record.

5.6.2 Fields

TABLE

Optional. The table name in current database.

If not supplied, the table name used is "ITEMS".

KEYFLDxx

Mandatory. The field name in the table used to search the record.

Each key identifier (KEYFLDxx) must be unique. The "xx" can have any value. The order of value identifiers is significant and must be the same as VALPOSnn.

VALPOSxx

Mandatory. The zero based data columns (in following data flow) used to search the record in the database.

Each value identifier (VALPOSxx) must be unique. The "xx" can have any value. The order of value identifiers is significant and must be the same as KEYFLDxx.

FIELDxx

Optional. Field identifier.

If field names are supplied, they must be the same quantity as columns in each row.

Each field identifier (FIELDxx) must be unique. The "xx" can have any value. The order of field identifiers is significant and is relative to data supplied.

5.6.3 Data

The data to fill the database fields, separated by "|" (pipe character).

If the field names are not supplied, the data is set from columns into fields in the same order of the table definition.

The number of data columns should be the same of fields in the table.

5.6.4 Usage

<DBLFILE>

<DATABASE>

<DATA-REPLACE TABLE="ITEMS" KEYFLD00="REF" VALPOS00="1" FIELD00="NAM" FIELD01="REF" FIELD03="OTHER">

MyNam1|001|Other1

MyNam2|002|Other2

</DATA-REPLACE>

</DATABASE>

<DBLFILE>

5.7 DATA-DELETE element tag

5.7.1 Description

Data to delete from a table in the current database.

The field(s) name(s) in the database denoted by KEYFLD is used to search the record(s). One or more fields are allowed, (with separators) to search in the database.

5.7.2 Fields

TABLE

Optional. The table name in current database.

If not supplied, the table name used is "ITEMS".

KEYFLD or KEYFLDnn

Mandatory. The field(s) name(s) in the database used to search the record.

5.7.3 Data

The data to search for records to delete in the database. If duplicated keys exists in database, all the records responding to the search are deleted.

One or more data fields will be supplied, separated by "|" (pipe character).

The number of data columns must be the same of KEYFLDnn.

5.7.4 Usage

<DBLFILE>

```
<DATABASE>
<DATA-DELETE TABLE="ITEMS" KEYFLD="REF">
001
002
</DATA-DELETE>
<DATA-DELETE TABLE="ITEMS" KEYFLD01="REF" KEYFLD01="NAM">
003|X
004|Y
</DATA-DELETE>
</DATABASE>
<DBLFILE>
```

6 Reserved keywords tags (further use)

6.1 VERSION element tag

6.1.1 Description

This element will be used for incremental update implementation.

6.1.2 Fields

NUM

Data flow version number.

6.1.3 Data

6.1.4 Usage

```
<VERSION NUM="123">
</VERSION>
```

6.2 OPTIONS element tag

6.2.1 Description

Set several options for parsing the DBL file.

6.2.2 Fields

ESCAPE-CTRL=YES

Option escape control chars activated.

FIELD=5E

Field separator character definition (in hex).

6.2.3 Data

6.2.4 Usage

```
<OPTIONS ESCAPE-CTRL="YES" FIELD-SEP="5E">
</OPTIONS>
```

End of option element

7 Samples

```
<DBLFILE>
<VERSION NUM="123"></VERSION>
<DATABASE PATH="//windows\wm\mychannel\mybase.sdf">
<SQL>
SELECT * FROM ITEMS</SQL>
<SQL> DELETE FROM PRODS</SQL>
<DATA-REPLACE TABLE="ITEM" KEYFLD="REF" VALPOS="0" FIELD01="REF"
FIELD02="NAM" FIELD03="OTHER">
0001|XXXXXXXXXXXXXXXXX|DDDDDDDDDD
0002|YYYYYYYYYYYYYYYYYYY|DDDDDDDDDD
0003|ZZZZZZZZZZZZZZZ|DDDDDDDDDD
</DATA-REPLACE>
<DATA-ADD TABLE="ITEM" FIELD01="REF" FIELD02="NAM">
0001| ZZZZZZZZZZZZZZ
0002|YYYYYYYYYYYYYYYYYYY
0003|ZZZZZZZZZZZZZZZ
</DATA-ADD>
<DATA-DELETE TABLE="ITEM" KEYFLD="REF">
0004
0005
0006
</DATA-DELETE>
<DATA-DELETE TABLE="ITEM" KEYFLD01="REF" KEYFLD01="NAM">
0004| ZZZZZZZZZZZZZZ
0005|YYYYYYYYYYYYYYYYYYY
0006|ZZZZZZZZZZZZZZZ
</DATA-DELETE>
</DATABASE>
</DBLFILE>
```

```

<DBLFILE>
<VERSION NUM="123"></VERSION>
<DATABASE>
<SQL>DROP TABLE TARIIFS</SQL>
<SQL>CREATE TABLE TARIIFS ( codtar ntext, lib ntext, codtarGRP ntext,
pxnet ntext)</SQL>
<DATA-ADD TABLE="TARIIFS" FIELD01="codtar" FIELD02="lib"
FIELD03="codtarGRP" FIELD04="pxnet">
11|PROMO EN PRESENTOIRS| |N
13|OP PROMO SEPT-OCT | |O
17|PRESENTOIRS RDC | |O
CU|RDC UNIVERSITAIRE |CN|N
M1|PROMO MATCH 03/04/20|CM|O
</DATA-ADD>
<SQL>DROP TABLE CLIENTS</SQL>
<SQL>CREATE TABLE CLIENTS ( codcli integer PRIMARY KEY, nomcli ntext,
codetar1 ntext, codetar2 ntext, codetar3 ntext, codetar4 ntext,
codetar5 ntext, codetar6 ntext, codetar7 ntext, codetar8 ntext,
codetar9 ntext, codetar10 ntext, rem ntext, etatcpt ntext, delai_liv
integer)</SQL>
<DATA-ADD TABLE="CLIENTS" FIELD01="codcli" FIELD02="nomcli"
FIELD03="codetar1" FIELD04="codetar2" FIELD05="codetar3"
FIELD06="codetar4" FIELD07="codetar5" FIELD08="codetar6"
FIELD09="codetar7" FIELD10="codetar8" FIELD11="codetar9"
FIELD12="codetar10" FIELD13="rem" FIELD14="etatcpt"
FIELD15="delai_liv">
008003|CHAMPION VLENNE |CN|CS|13|64|26| | | | | |12|N|02
960273|CASINO LYON MERMOZ |GC| | | | | | | | | |12|N|03
960279|HYPER U ROMANS SUR I|CN|CS|13|64|26| | | | | |00|N|03
960283|CASINO SUSVILLE |GC| | | | | | | | | |12|N|02
960333|CASINO VAUX EN VELIN|GC| | | | | | | | | |12|N|03
960336|CASINO LYON GERLAND |GC| | | | | | | | | |12|N|03
960406|CASINO GRENOBLE |GC| | | | | | | | | |12|N|02
</DATA-ADD>
<DATA-UPDATE TABLE="CLIENTS" FIELD01="delai_liv" KEYFLD01="codcli">
02|008003
03|960273
</DATA-UPDATE>
<SQL>DROP TABLE ARTICLES</SQL>
<SQL>CREATE TABLE ARTICLES ( codeart integer PRIMARY KEY, lib ntext,
gencod ntext, colisage integer)</SQL>
<DATA-REPLACE TABLE="ARTICLES" KEYFLD00="codeart" VALPOS00="0"
FIELD01="codeart" FIELD02="lib" FIELD03="gencod" FIELD04="colisage">
00001|4 BOITES ARCHIVES D8|3249441284090|005
00002|1 CAH PIQUE 17X22 96|3194441283102|010
07170|1 GOMME BLANCHE CAOU|3222472364535|030
07194|1 RAMETTE 500F A4 80|3222471224274|005
07201|12 FTRES DESSIN BTE |3222472225621|024
07202|10 CRAY COUL GROS MO|3222472225560|024
</DATA-REPLACE>
<SQL>DROP TABLE PRIX</SQL>
<SQL>CREATE TABLE PRIX ( codtar ntext, codart integer, pxht
integer)</SQL>
<DATA-ADD TABLE="PRIX" FIELD01="codtar" FIELD02="codart"
FIELD03="pxht">

```

```
13|06411|104964  
13|07051|014030  
GC|07193|000047  
GC|07194|000369  
GC|07195|000409  
GC|07196|000362  
GC|07201|000267  
GC|07202|000115  
</DATA-ADD>  
</DATABASE>  
</DBLFILE>
```