

## **FileMaker records to XML or JSON**

Sometimes you need to query FileMaker records as XML or JSON data and include related records. This can be tricky to do in pure FileMaker scripts and calculations, but via [MBS FileMaker Plugin](#) we can provide help. Our [FM.SQL.Execute](#) function can use SQL to fetch records. The [FM.SQL.CSV](#), [FM.SQL.JSONRecord](#) and [FM.SQL.JSONRecords](#) functions can help to pack those in comma/tab separated text or JSON. New for next plugin version are [FM.SQL.XMLRecord](#) and [FM.SQL.XMLRecords](#) functions to do for XML what we had for JSON already.

Below we have a sample script to use [FM.SQL.XMLRecords](#) to get records from two tables and insert the related records in the right position in the XML with our [XML.SetPathXML](#) function.

### **# SQL XML in file Contacts**

# Run query to fetch some records

```
Set Variable [ $sql1 ; Value: MBS( "FM.SQL.Execute"; "" ; "SELECT\n\"PrimaryKey\\", \"First Name\\", \"Last Name\\", \"Company\\", \"Title\\\"\nFROM Contacts" ) ]
```

# get them as XML

```
Set Variable [ $xml ; Value: MBS( "FM.SQL.XMLRecords"; $sql1;\n"people"; "person"; "ID¶First¶Last¶Company¶Title"; 2+1 ) ]
```

#

# loop over records to look for related record

```
Set Variable [ $count ; Value: MBS( "FM.SQL.RowCount"; $sql1 ) ]
```

```
Set Variable [ $index ; Value: 0 ]
```

```
If [ $index ≤ $count ]
```

```
    Loop
```

```
        # get primary key from result
```

```
        Set Variable [ $key ; Value: MBS( "FM.SQL.Field"; $sql1;\n$index; 0 ) ]
```

```
        # Run query for related records
```

```
        Set Variable [ $sql2 ; Value: MBS( "FM.SQL.Execute"; "" ;\n"SELECT Type, Number FROM \"Phone Numbers\" WHERE ForeignKey =\n?"; $key) ]
```

```
        # get them as XML
```

```
        Set Variable [ $xml2 ; Value: MBS( "FM.SQL.XMLRecords";\n$sql2; "phones"; "phone"; "Type¶Number"; 2+1 ) ]
```

```
        # Now insert them in the person node as new child phones
```

```
        Set Variable [ $xml ; Value: MBS( "XML.SetPathXML"; $xml;\n"people.person[" & $index &"].phones"; 0; $xml2 ) ]
```

```
        # don't forget to release memory
```

```
        Set Variable [ $r ; Value: MBS( "FM.SQL.Release"; $sql2) ]
```

```
#
# next
Set Variable [ $index ; Value: $index + 1 ]
Exit Loop If [ $index ≥ $count ]
End Loop
End If
#
# don't forget to release memory
Set Variable [ $r ; Value: MBS( "FM.SQL.Release"; $sql1) ]
# Output the XML with format & color
Set Field [ Contacts::XML ; MBS("XML.Colorize"; MBS("XML.Format";
$xml)) ]
```

We hope this helps people interested in XML exports. Same technique can be used with our [JSON](#) functions to build similar JSON structures.

**PS:** Script text copied with color using our copy button in Script Workspace.