

# Generic full data export/import routine

By Thomas Maul, General Manager, 4D Germany.  
TN 06-07

## Introduction

---

This Tech Note introduces a generic 4D method which exports or imports a whole database in XML format, including pictures, BLOBs and subtables.

A full text export/import always proves useful for long-term archiving and to repair damaged data files. Exporting all data in text only mode and re-importing into a newly created data file guarantees a clean data and indexes. Besides, it produces a perfectly defragmented data file, which increases sequential access speed.

The sequence number for each table is automatically restored.

All the code is in a single method, this allows easy installation (copy&paste) into existing structures.

## Installation

---

Copy the method "ExportImport" either from the example database (which contains only this method) or from text file "ExportImport.rtf" (which can be opened with most text editors on all platforms).

Create a method named "ExportImport". The name must be exactly "ExportImport", if you want to use another name you need to modify the method to reflect that change. Paste the contents of the clipboard.

If you compile using "All variables are typed" you need to copy the following lines into a Compiler\_xxx method:

```
C_TEXT(ExportImport ;$0;$1;$2)
C_POINTER(ExportImport ;$3)
C_LONGINT(ExportImport ;$4)
C_BOOLEAN(EXPORTIMPORT_STOP)
```

The method uses only one process variable, "ExportImport\_Stop". The variable can be used to stop an export/import, by example from an ON EVENT CALL method.

## Usage

---

The method can be directly executed from the User mode. If it is called without any parameter; it will open a window to ask if you want to export or import; then it will request to select an export/import folder.

During the export it creates one file for each table. Even if 4D can handle files larger than 2 GB on Windows XP, it may introduce problems on Mac OS, so the code automatically segments the files and starts additional files at 1.5 GB. The file is in XML format, pictures and blob fields are encoded as Base64.

### Note for the import:

The routine automatically deletes existing records and then start the import. It is supposed to be used in an empty (newly created) data file. Existing indexes are removed before importing and are then recreated. This improves the import speed and helps to have records and index pages with little fragmentation. The sequence number is automatically restored.

Two parameters allow you to integrate the method in your own user interface:

Result := ExportImport(direction; path)

Parameter	Type	Description
Direction	String	Import or Export
Path	String	path to export/import folder

Example:

```
$result := ExportImport ("Export";"C:\\myxmlfolder")  
$result := ExportImport ("Import";"C:\\myxmlfolder")
```

## Table/Field names

---

The method automatically converts table and field names to XML conform tag names. Diacritical characters are replaced with "\_", similar field names with leading digits.

Example:

```
Straße    ->   Stra_e  
2address  ->   _2address
```

## History

---

Tech Note 01-43 described a similar concept using components. Using 4D 6.7 the code was more complex and more difficult to install. Furthermore this new version supports automatically subtables and sequence numbers.