



# **i4XML Manual**

## **Programmer's Guide & Reference Manual**

Copyright GOERING iSeries Solutions

# Table of Contents

|  |           |
|--|-----------|
| <b>Part 1 Editorial/ Introduction</b>                  | <b>1</b>  |
| 1.1 Editorial .....                                    | 1         |
| 1.2 Introduction .....                                 | 3         |
| 1.3 Structure Of The Manual .....                      | 5         |
| 1.4 Release Notes .....                                | 6         |
| <b>Part 2 Installation</b>                             | <b>7</b>  |
| 2.1 Overview .....                                     | 7         |
| 2.2 Downloading From The Internet .....                | 8         |
| 2.3 Preparations .....                                 | 9         |
| 2.4 Update Proceedings .....                           | 10        |
| 2.5 Comfortable Method (BRAVO) .....                   | 11        |
| 2.6 INSTALL AID Variation .....                        | 12        |
| 2.7 Installation - manually transferred by FTP .....   | 13        |
| 2.8 Licensing .....                                    | 15        |
| <b>Part 3 Starting i4XML</b>                           | <b>16</b> |
| 3.1 Preparations .....                                 | 16        |
| 3.2 Starting i4XML .....                               | 17        |
| <b>Part 4 Generator</b>                                | <b>18</b> |
| 4.1 Overview .....                                     | 18        |
| 4.2 Extended Mode .....                                | 20        |
| 4.3 Example .....                                      | 21        |
| Generate the XML Extraction Script - Interactive ..... | 21        |
| Generate the XML Extraction Script - Manually .....    | 23        |
| Selection Of The Break Fields .....                    | 24        |
| Definition Of The XML Tags .....                       | 25        |
| Extract XML Data .....                                 | 27        |
| 4.4 Example: Complex Structure .....                   | 28        |
| Definition Of A Structure .....                        | 28        |
| Generation Of The XML Extraction Script .....          | 30        |
| Selection Of The XML Data .....                        | 33        |
| 4.5 1-Step-Mode (Fastpath) .....                       | 34        |
| <b>Part 5 Parser</b>                                   | <b>35</b> |
| 5.1 Overview .....                                     | 35        |
| 5.2 Parser - Interactive XML Data Analysis .....       | 36        |

|  |           |
|--|-----------|
| 5.3 XML Example File For The Parser .....                      | 37        |
| 5.4 Example .....  | 38        |
| Startup Parser .....   | 38        |
| Overview Of The XML Level .....                                | 39        |
| Editing The Single Levels .....                                | 40        |
| Confirmation Of The Key-Fields .....                           | 41        |
| Specify Output Files .....                                     | 42        |
| Creation Of The Physical Files .....                           | 43        |
| Attribution Of The XML Tags .....                              | 44        |
| Save Mapping And Populate Files .....                          | 45        |
| 5.5 Parser - Batch Processing .....                            | 46        |
| <b>Part 6 The Commands - Overview</b> .....                    | <b>47</b> |
| 6.1 GENXMLESC .....  | 48        |
| 6.2 EXTXMLDTA .....  | 51        |
| 6.3 GENXML .....   | 56        |
| 6.4 PARSEXML .....   | 61        |
| 6.5 PARSEXMLB .....  | 65        |
| 6.6 WRKPARSE .....   | 68        |
| 6.7 WRKHRY .....   | 70        |
| 6.8 EVALXML .....  | 71        |
| 6.9 MULTIPARSE .....   | 72        |
| 6.10 CPYPRSMAP .....   | 75        |
| 6.11 RSTPRSMAP .....   | 76        |
| 6.12 WRKGENMAP .....   | 77        |
| 6.13 MERGEXML .....  | 78        |
| <b>Part 7 Description of the Screens -<br/>Generator</b> ..... | <b>79</b> |
| 7.1 Overview .....   | 79        |
| 7.2 Select Break Fields .....                                  | 80        |
| 7.3 Field to XML Tag Mapping .....                             | 82        |
| 7.4 Mapping Name Screen .....                                  | 84        |
| 7.5 Mapping List .....   | 85        |
| 7.6 Hierarchies Definition .....                               | 86        |
| 7.7 Hierarchy List .....                                       | 87        |
| 7.8 Generator Mapping List .....                               | 88        |
| 7.9 Change Input File Name .....                               | 89        |
| <b>Part 8 Description of the Screens -<br/>Parser</b> .....    | <b>90</b> |
| 8.1 Overview .....   | 90        |
| 8.2 Confirm XML Levels .....                                   | 91        |

|               |                                   |            |
|---------------|-----------------------------------|------------|
| 8.3           | Split Level Details .....         | 93         |
| 8.4           | Next Level Details .....          | 95         |
| 8.5           | Element Data .....                | 96         |
| 8.6           | Specify Output Files .....        | 97         |
| 8.7           | Confirm File Specifications ..... | 99         |
| 8.8           | XML Tag to Field Mapping .....    | 100        |
| 8.9           | Select Fields .....               | 102        |
| 8.10          | Mapping List .....                | 103        |
| 8.11          | Parser Mapping List .....         | 104        |
| <b>Part 9</b> | <b>Other i4XML tasks</b>          | <b>105</b> |
|               | <b>Index</b>                      | <b>106</b> |

# 1 Editorial/ Introduction

## 1.1 Editorial



# ***i4XML***

## Manual

Edition "11/2007"

This edition applies to the licensed program *i4XML* Version 07/07 and to all subsequent releases and modifications until otherwise indicated in new editions.

Please verify that you are using the latest edition for this product. Go to <http://www.goering.us> for the latest version.

This document contains material that is protected by international copyright, trademark and other intellectual property laws. Unless otherwise specified, this document is intended for your licensed use only as covered by the Goering iSeries Solutions User License. You may not modify, copy, reproduce, republish, upload, post, transmit or distribute in any way any material, including code and software, from this document.

Editor:

**GOERING iSeries Solutions**  
Alter Unteröwisheimer Weg 19  
76646 Bruchsal  
Germany

phone: +49 (0)7251-9895-12  
fax: +49 (0)7251-9895-13  
e-mail: [info@goering.us](mailto:info@goering.us)  
web: <http://www.goering.us>

© Copyright **GOERING iSeries Solutions, 2007**

We reserve the right to change this text.

## 1.2 Introduction

Just as EDI (Electronic Data Interchange) has been a means to standardize the data exchanged between businesses in the past, eXtensible Markup Language (XML) is now grabbing the limelight.

*i4XML* is the program, which helps you to convert XML data to AS/400 files and vice versa:

- The [Generator](#) function of the tool simplifies the creation of XML data from existing AS/400 files
- The [Parser](#) function of the tool simplifies the storage of XML data to AS/400 files

### Generator Usage:

The Tool **Generator** is able to

- pick up data from AS/400 Database Files and create a XML Document in the IFS (Integrated File System) of the AS/400 System
- extract selected records and fields from AS/400 data files
- extract data from multiple files
- map database field names with user-defined XML tags
- store the Data Extraction Logic in XML/ Database file so that it can be reused
- run in Batch Mode (without any online screens)

### Parser Usage:

The Tool **Parser** is able to

- pick up data from XML Documents from IFS and write into AS/400 Database Files
- extract data from XML Documents and display it on the screen
- to map data from XML into existing Database File or New Database Files, In case of New Files, the Tool recommends a Database Layout but the user has the flexibility to customize the layout.

- provide the flexibility to map XML Tags with Database File Fields
- store the Data Extraction Logic (Mapping) in XML/ Database File, so that it can be reused
- parse and extract data from well-formed XML Documents, which may or may not have XML Schema
- run in Batch Mode (without any online screens)

## 1.3 Structure Of The Manual

1. [Installation](#)
2. [Calling the i4XML-Main Menu](#)
3. [The Generator - incl. detailed example](#)
4. [The Parser - incl. detailed example](#)
5. [The commands](#)
6. Description of the Screens for the [- Generator](#)  
[- Parser](#)

## 1.4 Release Notes

## 2 Installation

### 2.1 Overview

This Chapter describes how to:

- [Download](#) *i4XML*'s latest release from the internet.
- [Install](#) *i4XML* on your System.
- [Update](#) *i4XML* on your System
- [License](#) *i4XML* on your System.

## 2.2 Downloading From The Internet

*i4XML* can be downloaded from the internet at <http://www.goering.de>. The file contains a compressed AS/400 - and related iSeries-Backup-File with all necessary objects for running *i4XML*.

There are 3 ways to install *i4XML*:

- a) Using the [BRAVO Reader](#)
- b) Installation of the SAVF using the [INSTALL AID](#) from Bugbusters
- c) Installation of the SAVF using [FTP](#)

Information about the latest *i4XML* release level can be found at <http://www.goering.us>

## 2.3 Preparations

### 1. Extract the (.zip) Download-File

If you do not already have a utility to unzip archived files, we recommend the application Winzip, which can be downloaded from the Internet <http://www.winzip.com>.

After the extraction, you will have either (separate from the execution) the file I4XML.4ZS or (using the FTP variant) I4XML.SAV.

### 2. Verify that the FTP-Service on your AS/400 i.e. iSeries is active or start it.

Use the AS/400-command: STRTCPSVR SERVER(\*FTP) to start the service

### 3. Update

If you want an update for your existing *i4XML* installation, please rename your current library I4XML to I4XMLOLD.

Use the AS/400-command:  
RNMOBJ OBJ(I4XML) OBJTYPE(\*LIB) NEWOBJ(I4XMLOLD)

The installation program recognizes this library and adopts all the settings from it.

## 2.4 Update Proceedings

### 1. Preparations:

If you want to load the latest release, please rename an already existing library before the upload:

I4XML to I4XMLOLD

Please use the AS/400-command:

```
RNMOBJ OBJ(I4XML) OBJTYPE(*LIB) NEWOBJ(I4XMLOLD)
```

That way the settings will be restored to the new version.

### 2. Installation:

Please carry the installation out, according to the desired method, as described in the following pages.

### 3. Assumptions on the settings

The setup program (called CALL I4XMLSETUP) realizes that there is already an older version installed and offers the assumption of the settings with the function key **F10**.

## 2.5 Comfortable Method (BRAVO)

If you have selected the method (BRAVO) of downloading, you will need the software ZipSeries for the installation.

This software tool can be downloaded from the following link  
<http://www.goering.de/download/ZipSeriesSetup.zip>

Please proceed to the chapter [Licensing](#).

## 2.6 INSTALL AID Variation

To easily transfer *i4XML* to your System AS/400, we also recommend using the tool INSTALL AID from Bugbusters.

The tool can be downloaded from <http://www.bugbusters.net> (It is a free download)

After the installation on your PC, you have to open the file I4XML.SAV. Please fill out the dialogueboxes with normal values.

Please proceed to the chapter [Licensing](#).

## 2.7 Installation - manually transferred by FTP

### 1. Create the backup-file on your System

Use the name i4XML and the AS/400-command:  
CRTSAVF FILE (QGPL / i4XML) AUT(\*ALL)

Open a DOS Window on your PC and change to the subdirectory which contains the ASCII file I4XML.SAV ([see chapter Installation-Preparations](#)).

### 2. Sign on to your System AS/400 and accordingly iSeries.

\*Note that the name "AS400" will be used for the name of your system AS/400 and accordingly iSeries and the color blue for all responding messages from the FTP.

```
FTP AS400
Connected with AS400
220-QTCP at AS400.FIRMA.DE
220 Connection will close if idle more than 5 minutes.
User (AS400: (none)): USER
331 Enter Password
Password PASSWORD
Note that neither the password will be shown nor the cursor will move
during entry.
230 USER logged on.
```

### 3. Change into the binary mode

```
FTP> binary
200 Representation type is binary IMAGE.
```

### 4. Change the naming format

```
FTP> quote site namefmt 1
250 Now using naming format "1".
```

### 5. Change to the library QGPL of your System AS/400 and accordingly iSeries

```
FTP> cd_/qsys.lib/qgpl.lib
250 "QSYS.LIB/QGPL.LIB is current library.
```

## 6. Start the transfer from PC to the AS/400 and accordingly iSeries

```
FTP> put I4XML.SAV i4XML.SAVF (replace)
200 PORT subcommand request successful.
150 Sending file to member I4XML.SAV in file
i4XML.SAVF in library QGPL.
Note that the transfer time will take a few minutes.
250 File transfer completed successfully.
FTP 123456789 Bytes sent in 98.76 seconds 54.321
KB/Sec.
```

## 7. Terminate the FTP-Session on the System AS/400 and accordingly iSeries.

```
FTP> quit
221 QUIT subcommand received.
```

## 8. Close the DOS command entry window.

The transfer to your System AS/400 i.e. iSeries is complete.

## 9. Create the *i4XML*-Objects on the System AS/400 and accordingly iSeries

Using the AS/400-command:

```
RSTLIB SAVLIB (I4XML) DEV (*SAVF) SAVF (QGPL/i4XML)
```

Please proceed to the chapter [Licensing](#).

## 2.8 Licensing

### 1. Add the library to your Library Search list in your Session

Use the AS/400-command: `ADDLIBLE I4XML`

Installation of *i4XML* is now complete and, after entering the license key, ready for execution.

### 2. Call the licensing for *i4XML*

Use the command: `CALL I4XMLCODE`

### 3. Note the system specifications of your AS/400.

The *i4XML* licensing display will indicate the

- a) Serial-No.,
- b) Model-No. and
- c) Processor-No.

of your AS/400 or iSeries. If you don't already have a license key, please send the system information in an e-mail to: [info@goering.de](mailto:info@goering.de). You will receive your *i4XML* license key within 24 hours via e-mail.

The quickest way to get your license code (to receive a license within 30 minutes!) is to input your system specifications on the Download web page at <http://www.goering.de>. If you fail to enter the system specifications when downloading, you can revisit the page, enter the system information and select the button "Code Only".

### 4. Enter your *i4XML*-license key

Use the command: `CALL I4XMLCODE`

Enter the license key and press the Enter Key, *i4XML* is now ready to run. Please note that the program tolerates only three repeated attempts of false license keys entries per day. Further attempts will cause a deactivation of the *i4XML* licensing for the rest of each day.

## 3 Starting i4XML

### 3.1 Preparations

Please create a directory /I4XML (directly under the root) in the IFS of the AS/400 with the command CRTDIR '/I4XML'

This directory should be made to the current directory with the command: CHGCURDIR '/I4XML'.

If you are familiar with the IFS, you can also use other paths.

We can now start the program i4XML with [the start-up command GO I4XML](#).

## 3.2 Starting i4XML

After starting the program i4XML with the command GO I4XML, the following main menu is displayed:

---

```
i4XML                               the iSeries XML Tool!

Select one of the following:

    ** Do you want ... to create relational XML from your iSeries data?
1. i4XML Generator

    ** Do you want ... to load relational XML data into your iSeries?
2. i4XML Parser

    ** Looking for administrative functions, SDK/samples and others?
50. other i4XML tasks

90. Sign off
Selection or command
===>

    F3=Exit           F4=Prompt.         F9=Retrieve         F12=Cancel
    F13=Information Assistant  F16=AS/400-Main Menu
```

---

## 4 Generator

### 4.1 Overview

After pressing **option 1** of the i4XML Main Menu the Generator Main Menu will be started:

---

```

i4XMLG                               the iSeries XML Tool!

Select one of the following:

    ** Generator/one-step **
  1. Generate XML in Fastpath

    ** Generator/extended **
 10. Generate XML Extraction Script
 11. Extract XML data

    ** Additional tools **
 50. Work with Hierarchies
 51. Work with Mappings

    90. Sign off
Selection or command
====>

    F3=Exit           F4=Prompt.         F9=Retrieve         F12=Cancel
    F13=Information Assistant   F16=AS/400-Main Menu

```

---

For creating XML files from AS/400 files, there are strictly 2 alternatives:

#### 1. The Extended Mode (Standard)

To begin, an extraction script is created with the command [GENXMLESC](#) (**Option 10**), which contains the complete definition for the later generation of XML Data. After this, the Data can be [analyzed and generated](#) via the extraction script (**Option 11**).

This extraction script can be used multiple times for generation of XML Data. I4XML can then be started in each case with the command [EXTXMLDTA](#)

[Complex structures](#) can also be handled. Use the command [WRKHRY](#) before the Generation of the Extraction Script, to define the Structure of the file to be generated. In this case when executing [GENXMLESC](#) you only need to refer to the created Hierarchy.

More information can found in the chapter "[Extended Mode - Proceeding with structures of a high complexity](#)" and "[Example: Complex Structure](#)"

## [2. The 1-Step-Mode \(Fastpath\)](#)

This alternative interfaces the creation of an Extraction Script and the Data Extraction and is initiated by the command [GENXML](#) (**Option 1**).

## 4.2 Extended Mode

Proceeding with structures of a normal complexity:



Fig. samplesimple.jpg

At first, an Extraction Script will be created with the command [GENXMLESC](#) (one-time action), which contains the complete definition for the later generation of XML Data.

After this, the Data is [analyzed and generated](#) by means of the Extraction Script.

This Extraction Script can be used multiple times for generation of a XML file. I4XML is then started with the command [EXTXMLDTA](#)

Proceeding with structures of a high complexity:



Fig. samplecomplex.jpg

The first move is to define a hierarchy with the command [WRKHRY](#) (**Option 50 Generator-Main menu**) - q.v. chapter "[Example: Complex Structure](#)"

The further steps are comparable with the definition of normal complex structures.

**Note!** Several screens can be different in this case!

## 4.3 Example

### 4.3.1 Generate the XML Extraction Script - Interactive

By means of an example, we will show you the creation of an Extraction Script. (The alternative is a [manually written Extraction Script](#), which will be uploaded to the IFS of the AS/400 system.)

A XML Extraction Script contains all instructions which are needed to create a XML-Output. The File is stored in the IFS (Integrated File System) and (optionally) in an AS/400-physical file (Mapping).

If you choose **option 10** "Generate XML Extraction Script" in the [Generator Main Menu](#) , the following command will be processed:

```
GENXMLESC SCRIPT (file2xml.esc) INPUTTYPE(*FILE) FILENAME  
(YCUSTOMER)
```

The complete parameter description is described in the chapter: [The Command GENXMLESC](#).

Because we didn't declare a mapping at the start-up, the program goes into the dialogue mode.

At first, an Extraction Script is created, which contains all information for the generation of the XML Data. This information is saved in a XML File in the IFS (Integrated File System) and in an AS/400 Database File. The XML Extraction Script is created in a XML Standard format.

If the Extraction Script is to be created interactively, the following display "Generate XML Extraction Script" will be appearing. You have to enter the "XML Script Name" and in the field "File Name" the data source, on which the XML file will be created.

Depending on the source of the data you have to input the adequate type in the field "Input Object Type": "\*QRYDFN" (an already existing query), "\*FILE" (a file) and "\*MAPQRY" (first creating a query or using an already existing MAPQRY).

In our example, we create an Extraction Script with the name FILE2XML, the data source is YCUSTOMER.

---

```

Generate XML Extraction Script (GENXMLESC)

Type choices, press Enter

XML Script Name   :   FILE2XML.ESC
_____
_____

Input Object Type:   *FILE_____   *QRYDFN, *FILE, *MAPQRY

                                                    Bottom
F3=Exit  F4=Prompt F5=Refresh  F12=Cancel F13=How to use this display
F24=More Keys

```

---

After pressing the **Enter Key**, we have to input in the following display the filename (YCUSTOMER) and the library (I4XML).

---

```

Generate XML Extraction Script (GENXMLESC)

Type choices, press Enter

XML Script Name   :   'FILE2XML.ESC'
_____
_____

Input Object Type . . . . . *FILE_____   *QRYDFN, *FILE, *MAPQRY
File Name . . . . . YCUSTOMER_____   Name
Library . . . . . I4XML_____   Name, *LIBL
XML Hierarchy Definition Name. *NONE
XML Tag Mapping Name. . . . . *SELECT   Name, *SELECT, *DEFAULT
Replace. . . . . *YES   *YES, *NO

                                                    End
F3=Exit  F4=Prompt F5=Refresh  F12=Cancel F13=How to use this display
F24=More Keys

```

---

After the reconfirmation we will go to [the Selection of the Breakfields](#)

### 4.3.2 Generate the XML Extraction Script - Manually

If the XML Extraction Script is written manually, the following configuration should be considered:

```
<?xml version="1.0" ?>
<XMLESC>
<SQL>
```

The Query, which will create the output for the XML Data generation

```
< MAPPING>
  < MAP>
    <DBLIB> Library </DBLIB>
    <DBFILE> File </DBFILE>
    <DBFIELD> Field </DBFIELD>
    <TYPE> Datatype of Database Field </TYPE>
    <LEN> Field Length </LEN>
    <DEC> Number of Decimal Positions for Numeric Data Type </DEC>
    <TAG> XML Tag Name </TAG>
    <BREAKFIELD> 'Y' , if it is the first node of the tree, else 'N'
  </BREAKFIELD>
    <BREAKORDER> Level at which the tree starts, only applicable if
    BREAKFIELD is 'Y' </BREAKORDER>
    <BREAKTAG> Parent Tag Name, only applicable if BREAKFIELD is
    'Y' </BREAKTAG>
    <HIERARCHY> Hierarchy for this XML Tag </HIERARCHY >
    <ELEMTYPE> Type of the Tag (Element/ Attribute) </ELEMTYPE>
  </MAP>
</MAPPING>
</XMLESC>
```

### 4.3.3 Selection Of The Break Fields

In the column "Break Y/N" we set the field CUSTOMER as the break field (input "Y") and in the column "Break Lvl" we enter '001' for the level. The break fields will now be tagged with a (break) tag name. The parent tag for a break field is always the break tag of its immediate upper level.

---

```

I4XS012                                XML Generator                30.07.07
                                       Select Break Fields           17:37:26

Extraction Script Name : /I4XML/FILE2XML.ESC

File      Field      Break Break      Parent
Name      Name Y/X/N/A Lvl Tag Name  Tag Name
YCUSTOMER CUSTOMER  Y 001 YCU_CUSTOMER
YCUSTOMER NAME      N 000 YCU_NAME
YCUSTOMER CITY      N 000 YCU_CITY
YCUSTOMER STREET   N 000 YCU_STREET
YCUSTOMER STREET2  N 000 YCU_STREET2

End
F3=Exit F6=Confirm F8=Show SQL F10=Load Mapping F11=Display Data F12=Cancel

```

---

q.v. [Description of the Screens - Generator, Select Break Fields](#)

With the function key **F6(Confirm)** we confirm the settings and go to the next menu "[Field to XML Mapping](#)". In this part, we will [confirm the XML Tags](#).

If there already exists a mapping, you can display a [selective list with existing mappings](#) with the function key **F10(Load Mapping)**.

After the selection of a mapping and pressing the **Enter Key**, you return to the above menu and confirm the settings with the function key **F6(Confirm)**. Now you can [define the XML Tags](#).

#### 4.3.4 Definition Of The XML Tags

On the screen "[Field to XML Mapping](#)" the name of the XML tag, the tag type (E = Element, A = Attribute) and the level of the individual tags (the hierarchy within the XML file) needs to be defined.

In the following example, the tag names are written in lower-case for easier readability:

---

```

I4XS014                XML Generator                30.07.07
                        Field to XML Tag Mapping    17:48:35

Extraction Script Name : FILE2XML.ESC

File      Field      Type Len  Dec XML Tag Name      Tag Type Lvl
Hierarchy
YCUSTOMER CUSTOMER  A     7   Customer          E    002
/YCU_CUSTOMER
YCUSTOMER NAME      A    30   Name              E    002
/YCU_CUSTOMER
YCUSTOMER CITY      A    20   City              E    002
/YCU_CUSTOMER
YCUSTOMER STREET    A    20   Street            E    002
/YCU_CUSTOMER
YCUSTOMER STREET2   A    20   Street2           E    002
/YCU_CUSTOMER

                                                End
F3=Exit      F8=Show SQL  F9=Save Mapping and Generate Extraction Script
F11=Generate Extraction Script F12=Cancel

```

---

q.v. [Description of the Screens - Generator, Field to XML Tag Mapping](#)

After pressing the function key **F9(Save Mapping and Generate Extraction Script)** a small window "[Mapping Name Screen](#)" is displayed.

Here you can choose an existing mapping name, press function key **F4(List Mapping)**, or create a new mapping name (maximal length 10 characters, e.g. FILE2).

---

```

I4XW002    Mapping Name Screen

Entering Mapping Name: _____

F4=List Mapping    F12=Cancel

```

---

q.v. [Description of the Screens - Generator, Mapping Name Screen](#)

After the confirmation of the settings with the **Enter Key**, the mapping is created and saved. Return to the [Generator Main Menu](#) and select **option 11 "Extract XML Data"** for completing the real generation and extraction of the XML Data.

### 4.3.5 Extract XML Data

After selecting **option 11**, "Extract XML Data" from [the Generator Main Menu](#) (command [EXTXMLDTA](#)), the following screen is displayed:

---

```

                                Extract XML Data (EXTXMLDTA)

Enter your selection and press ENTER.

Path of 'XML Extractor Script'      /I4XML/FILE2XML.esc

Path of XML . . . . .                /i4XML/FILE2XML.xml

Generate DTD . . . . .                *NO          *INT, *EXT, *WEB, *NO
Transfer by FTP . . . . .            *NO          *YES, *NO
Replace File . . . . .                *YES         *YES, *NO

F3=Exit      F4=Prompt      F5=Refresh      F10=Other Parameters      End
F12=Cancel   F13=Information Assistant  F24=Other Keys

```

---

The complete parameter description is described in the chapter: [The Command EXTXMLDTA](#).

In the field "Path of 'XML Extractor Script'" we enter the name of the newly created Extraction Script (source)

In the field "Path of XML" we enter the name of the XML stream file (target).

Because the DTD shouldn't be created, the field "Generate DTD" should be set to to "\*NO".

The data of the source file is chosen like the specifications and the output file File2xml.xml is created in XML Structure. Afterwards, I4XML returns to the [Generator Main Menu](#)



For our example we will create a new hierarchy (**F6(Create)**). Now we will be asked for the name of the hierarchy, which we will give the name CUSTOMER

---

Create Hierarchy Definition

```
Hierarchy Definition . . . . . CUSTOMER
XML Name . . . . . *NONE
```

F12=Cancel

---

We confirm this and go on to the screen "[Hierarchies Definition](#)" where we can define the new structure. The "Hierarchy" is then later used when we create the "Extraction Script". This approach is recommended for large and complicated structures.

The levels "Customers" and "Customer\_Set" themselves are not a part of the structure, and they will be each ignored during the generation. So we will omit them during the generation.

---

```
I4XS020                                XML Generator                                27.08.07
                                      Hierarchies Definition                            11:15:37
Hierarchy Definition : CUSTOMER
Type options, press Enter.
  1=Insert    4=Delete

Opt   Order   Level   Break Tag
     10      1      Customer_Detail
     20      2      Id_n_other
     30      2      Name_n_other
     40      3      Streets

F3=Exit      F5=Reorder    F6=Save      F17=Reorganize

Weitere ...
```

---

q.v. [Description of the screens - Generator, Hierarchies Definition](#)

With the function key **F6(Save)** we leave the definition and for the future we can reference this hierarchy.

#### 4.4.2 Generation Of The XML Extraction Script

Like in the prior (simple) example we now generate an extraction script with the following command:

```
GENXMLESC SCRIPT(FILE2XMLSTRUCT.ESC) INPUTTYPE(*FILE)
FILENAME(YCUSTOMER)
HIRDEF(CUSTOMER)
```

The exact parameter description can be found in the chapter: [The Command GENXMLESC](#).

Because there is no mapping, stated in the program call, the program starts in the dialogue mode. The first step is to select the Break Fields at screen "Select Break Fields".

As opposed to the simple case there is no input option proffered for the "Break Tags", but you can choose the desired tags with the function key **F4(Hierarchy)** in the field "Lvl" from the hierarchy definition (q.v. screen "Hierarchies").

---

```
I4XS021                XML Generator                27.08.07
                        Select Break Fields          11:45:04

Extraction Script Name : FILE2XMLSTRUCT.ESC

File      Field      Break
Name      Name Y/X/N/A Lvl Hierarchy
YCUSTOMER CUSTOMER  Y 000
YCUSTOMER NAME       N 000
YCUSTOMER CITY       N 000
YCUSTOMER STREET    N 000
YCUSTOMER STREET2   N 000

                                                End
F3=Exit  F4=Hierarchy  F6=Confirm  F8=Show SQL  F10=Load Mapping
F11=Display Data  F12=Cancel
```

---

[q.v. Description of the Screens - Generator, Select Break Fields](#)



After all fields are processed the screen should look as follows:

---

```
I4XS021                                XML Generator                                31.07.07
                                         Select Break Fields                            17:48:59

Extraction Script Name : /I4XML/FILE2XMLSTRUCT.ESC

File      Field      Break
Name      Name Y/X/N/A Lvl Hierarchy
YCUSTOMER CUSTOMER  Y 002 /Customer_Detail/Id_n_other
YCUSTOMER NAME      Y 002 /Customer_Detail/Name_n_other
YCUSTOMER CITY      N 000
YCUSTOMER STREET    Y 003 /Customer_Detail/Name_n_other/Streets
YCUSTOMER STREET2   N 000

                                                                 End
F3=Exit  F4=Hierarchy F6=Confirm F8=Show SQL F10=Load Mapping
F11=Display Data F12=Cancel
```

---

q.v.. [Description of the Screens - Generator, Select Break Fields](#)

With the function key **F6(Confirm)** we proceed with the details definition i.e. with the screen "[Field to XML Tag Mapping](#)"

This screen is identical to the screen of the simple structured sample.

We will finish the generation with the function key **F9(Save Mapping and Generate Extraction Script)** and assign the mapping name "STRUCT".

### 4.4.3 Selection Of The XML Data

We start the generation with the following command:

```
EXTXMLDTA SCRIPT(FILE2XMLSTRUCT.ESC)
PATH(FILE2XMLSTRUCT.XML) MASTTAG(Customers)
RECLVLTAG(Customer_Set)
```

Both of the last parameters will be visible in the "**F4 mode**" by pressing the function key **F10**.

The exactly description of the parameter scan you read in the chapter: [The Command EXTMLDTA](#).

## 4.5 1-Step-Mode (Fastpath)

This option is used to generate a XML Data file in the fast path without using any prior created XML Extraction Script File.

The Fastpath

- join the two commands [GENXMLESC](#) and [EXTXMLDTA](#) to one command:  
GENXML
- can be call up both from the [Generator Main Menu](#) and from the command line with the following command:  
GENXML SCRIPT(FILE2XML.XML) OBJTYP(\*FILE) OBJNAM1(YCUSTOMER)

Parameter description:

|         |   |
|---------|---|
| SCRIPT  | = defines the XML file, which will be created |
| OBJTYP  | = the type of the source file                 |
| OBJNAM1 | = identified the source file                  |

The complete parameter description is described in the chapter: [The Command GENXML](#).

If you want to use an existing mapping, e.g. FILE2 (see chapter "[Generator - Definition Of The XML Tags](#)"), you have to add it to the command the parameter TAGNAM(FILE2).

## 5 Parser

### 5.1 Overview

After pressing the option 2 of the i4XML Main Menu the Parser Main Menu will be started:

---

```
I4XMLP                                the iSeries XML Tool!

Select one of the following:

    ** Parser                          **
  1. Parse XML File-interactive

 10. Parse XML File in Batch \(use this for larger files\)
 11. Work with Parsed Output

    ** Additional tools                **
 50. Work with Mappings
 51. Export Mapping
 52. Import Mapping

 70. Multi-Parser \(use this to full automate processing\)

 90. Sign off
Selection or command
===>

F3=Exit          F4=Prompt.          F9=Retrieve          F12=Cancel
F13=Information Assistant  F16=AS/400-Main Menu
```

---

We will differentiate between two types of XML Data Analysis:

1. The [Interactive XML Data Analysis \(Option 1\)](#), the import of the XML files will be done within any interaction.
2. The [Batch processing \(Option 10\)](#), is recommended for the conversion of large XML input files because this option will save your system resources and doesn't block your interactive session.

## 5.2 Parser - Interactive XML Data Analysis

We will demonstrate the functionality of the parser with a [detailed example](#). We will use the XML file "PARSERDEMO.XML", which is included in the installation package of i4XML. Please transfer this file to your AS/400 system via FTP or a "linked folder".

In parallel, please display the XML file on your PC with [Internet Explorer](#) or another browser.

In this example we will use a file with Customers' order and Offer data. We will create the following files:

- a) customer base
- b) order heading
- c) order positions
- d) offer heading
- e) offer positions

The following example is structured over several pages for simplicity of understanding.

### 5.3 XML Example File For The Parser

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <package>
- <customer>
  <customernr>0000001</customernr>
  <name>Best Company</name>
- <order>
  <ordernr>4711</ordernr>
  <value>123456,78</value>
  - <line>
    <pos>1</pos>
    <partnr>0815/2004</partnr>
  </line>
</order>
- <offer>
  <offernr>4712</offernr>
  <value>22222,99</value>
  - <line>
    <pos>11</pos>
    <partnr>2815-2004</partnr>
  </line>
</offer>
</customer>
</package>
```

## 5.4 Example

### 5.4.1 Startup Parser

We start the Parser with the command [PARSEXML](#) - **option 1** of the [Parser Main Menu](#).

In the following Screen we provide the XML file name with its path at field "XML Document Name" and enter \*CREATE at field "XML Tag Mapping Name". Another field of interest is "Decimal Format" where one can assign the used decimal point. The other default values can be left as they are.

If you enter \*SELECT at the field "XML Tag Mapping Name" another screen appears with the existing mappings. If you choose from there the processing will be done without any additional dialogue and then return to the [Parser Main Menu](#).

```

Parse XML Data-Interactive (PARSEXML)

Enter your selection and press ENTER .

XML Document Name . . . . . /HOME/GOERING/XML/MANUAL/PARSERDEMO.XML

Validate XML against DTD . . . . *NO          *YES, *NO
XML Tag Mapping Name . . . . . *CREATE     Name, *CREATE, *SELECT, *D...
Consider Merge/Split Mapping . . *YES       *YES, *NO
Ignore Error . . . . . *NO          *YES, *NO
Copy data into One file . . . . *NO        *YES, *NO
Convert Attributes to Element . . *YES       *YES, *NO
Decimal Format . . . . . ' ,'          ' '=Comma, '.'=Dot
Enter new File Name . . . . .          Name
Library . . . . .          Name

F3=Exit      F4=Prompt.      F5=Refresh      F12=Cancel      More ...
F13=Information Assistant      F24=Other Keys

```

The complete parameter description is described in the chapter: [The Command PARSEXML](#).

After confirmation of the input, the next screen shows an [overview of the XML Level](#) from the example file.

## 5.4.2 Overview Of The XML Level

In the "Confirm XML Levels" screen, the expected 5 levels are displayed. Now [edit the individual levels](#) for each parent tag. Put the cursor on the desired level and enter **option 5** "Next Level Details".

However, first edit the parent tag CUSTOMER:

Please choose the **option 5** "Next Level Details" in the row CUSTOMER and press the **Enter Key**.

---

```

I4XS001                                XML Parser                                2.08.07
                                       Confirm XML Levels                          12:26:45

XML File Name :    /HOME/GOERING/XML/MANUAL/PARSERDEMO.XML
Type options, press Enter.

2=Split Level      5=Next Level Details      6=Merge Level      7=Hierarchy

Opt  Parent                Level      Relation
     CUSTOMER              02         N
     ORDER                 03         N
     LINE                  04         N
     OFFER                 03         N
     LINE                  04         N

F2=Show XML      F3=Exit      F6=Confirm      F10=Load Mapping      F12=Cancel      End

```

---

q.v.. [Description of the Screens - Parser, Confirm XML Levels](#)

The next screen is "[Next Level Details](#)", which displays the lower levels of the parent tag CUSTOMER. [These levels can be edited](#).

### 5.4.3 Editing The Single Levels

In the screen "Next Level Details" the elements of a selected Parent Tag are shown.

Some of these Elements can assigned as "General Key", which means that all sublevels will get this field implemented too, although they don't have them in the original XML file.

In our samples we can see that the Parent CUSTOMER contains the Tags CUSTOMERNR and NAME:

---

```

I4XS003                XML Parser                2.08.07
                        Next Level Details        12:31:07

Hierarchy : /#document/PACKAGE/CUSTOMER
Parent   : CUSTOMER                        Level : 02

Elements                General Key (Y/N)

CUSTOMERNR              -
NAME                    -

F3=Exit      F6=Confirm  F10=Show Data  F12=Cancel      Ende

```

---

q.v.. [Description of the Screens - Parser, Next Level Details](#)

For displaying the values for NAME, move the cursor to the row NAME and press the function key **F10(Show Data)**:

---

```

I4XS005                XML Parser                2.08.07
                        Element Data            12:33:25

Hierarchy : /#document/PACKAGE/CUSTOMER
Parent   : CUSTOMER                        Level : 2
Element  : NAME

Row  Value
  1  Best Company

F3=Exit      F12=Cancel      End

```

---

q.v.. [Description of the Screens - Parser, Element Data](#)

After reading the values, we will return to the screen "[Next Level Details](#)" (see above) by pressing the function key **F12(Cancel)**, now we will [confirm the Key-fields](#).

#### 5.4.4 Confirmation Of The Key-Fields

An important feature of the parser is the confirmation of the Key-Fields. It is necessary in the following process to establish a context between the different levels so that the files will be properly generated and filled. In our example, we confirm the CUSTOMERNR as the "General Key". This means in all lower levels the CUSTOMERNR will be generated in the file.

In the screen "[Next Level Details](#)", enter "Y" in the column "General Key" for the element CUSTOMERNR

---

```

I4XS003                                XML Parser                                2.08.07
                                         Next Level Details                            12:36:34

Hierarchy : /#document/PACKAGE/CUSTOMER
Parent    : CUSTOMER                                Level : 02

Elements                                General Key (Y/N)

CUSTOMERNR                                Y
NAME                                         -

F3=Exit      F6=Confirm      F10=Show Data      F12=Cancel      End

```

---

q.v. [Description of the Screens - Parser, Next Level Details](#)

After confirming with function key **F6(Confirm)** we will return to the screen "[Confirm XML Levels](#)". Edit the parent tags ORDER and OFFER in the same way.

Using **option 5**, call the following screen "Next Level Details" and set the general key (= "Y") for the fields ORDERNR and OFFERNR. Confirm these settings with function key **F6(Confirm)**.

After the completion of all the settings, we will confirm them in the screen "[Confirm XML Levels](#)" with function key **F6(Confirm)**.

At present we have assigned the Key-Fields, next we will generate a DDS and create the physical files.

This will be carried out in the following screen "[Specify Output Files](#)" where you can enter the names of physical files according to the XML levels.

### 5.4.5 Specify Output Files

After the generation of the Key-Fields a DDS will be generated and the physical files will be created.

In the "[Specify Output Files](#)" screen we are entering the file information (written in red font), which is required to create / overwrite the files in the IFS of the AS/400:

---

```

I4XS007                                     2.08.07
                                           12:39:49

                               Specify Output Files

      XML File  /HOME/GOERING/XML/MANUAL/PARSERDEMO.XML

Lvl Parent Tag                               Library  File Name  File  Option  Replace
2  CUSTOMER                                QTEMP     CUSTOMER   Y    *ADD    N
3  ORDER                                    QTEMP     ORDER      Y    *ADD    N
4  LINE                                     QTEMP     ORDERLINE  Y    *ADD    N
3  OFFER                                    QTEMP     OFFER      Y    *ADD    N
4  LINE                                     QTEMP     OFFERLINE  Y    *ADD    N

                                           End

Press enter to continue
F2=Parent Hierarchy  F7=Element Details  F12=Cancel  F13=Repeat

```

---

q.v.s.a. [Description of the Screens - Parser, Specify Output Files](#)

After pressing the **Enter Key**, the confirmation screen "[Confirm File Specifications](#)" appears.

## 5.4.6 Creation Of The Physical Files

The "[Confirm File Specifications](#)" screen shows all files with their fields. The new generated tags are written in red font. The field lengths can still be changed here.

---

```

I4XS010                                XML Parser                                2.08.07
                                      Confirm File Specifications                12:48:50

XML Name    /HOME/GOERING/XML/MANUAL/PARSERDEMO.XML

Level  Parent Tag      Library      File      Parsed Tag      Field      Type Len  Dec  Datfmt
  2    CUSTOMER      QTEMP      CUSTOMER      CUSTOMERNR      CUSTOMERNR  N  007  0
  2    CUSTOMER      QTEMP      CUSTOMER      NAME            NAME        C  012  0
  3    ORDER        QTEMP      ORDER        CUSTOMERNR      CUSTOMERNR  N  007  0
  3    ORDER        QTEMP      ORDER        ORDERNR        ORDERNR     N  004  0
  3    ORDER        QTEMP      ORDER        VALUE          VALUE       N  008  2
  4    LINE         QTEMP      ORDERLINE    CUSTOMERNR      CUSTOMERNR  N  007  0
  4    LINE         QTEMP      ORDERLINE    ORDERNR        ORDERNR     N  004  0
                                          More ...
F2=Parent Hierarchy  F6=Confirm  F10=Display Data  F12=Cancel

```

---

q.v. [Description of the Screens - Parser, Confirm File Specifications](#)

After pressing the function key **F6(Confirm)**, the "[XML Tag to Field Mapping](#)" screen is displayed, and here we can [attribute the XML tags](#).

### 5.4.7 Attribution Of The XML Tags

In the "XML Tag to Field Mapping" screen we can see all necessary fields for creating the physical files.

Each tag of the XML file is displayed as well as its relation to the database field. The newly generated tags are displayed in red font.

If the files are generated by i4XML, the fields are filled out correctly. Otherwise the assignment must be done manually. To make this easier there is the Key **F4(Prompt)** which offers a comfortable [Selection screen](#). Tags, which you don't want to populate to a field, just assign with "\*OMIT"

In our sample we can see here all Fields that are needed to populate the file as desired:

---

```

I4XS006                                XML Parser                                2.08.07
                                         XML Tag to Field Mapping                    12:59:47

                                         XML File  /HOME/GOERING/XML/MANUAL/PARSERDEMO.XML

Level  Parent Tag      File      Parsed Tag      DB Field      ConvTyp      Type Flag
      Library
2      CUSTOMER        CUSTOMER  CUSTOMERNR      CUSTOMERNR    _____  N      Y
      QTEMP
2      CUSTOMER        CUSTOMER  NAME            NAME          _____  C      N
      QTEMP
3      ORDER          ORDER    CUSTOMERNR      CUSTOMERNR    _____  N      N
      QTEMP
3      ORDER          ORDER    ORDERNR         ORDERNR       _____  N      Y
      QTEMP
3      ORDER          ORDER    VALUE           VALUE         _____  C      N
      QTEMP
4      LINE           ORDERLINE  CUSTOMERNR      CUSTOMERNR    _____  N      N
      QTEMP

```

More ...

F2=Parent Hierarchy F3=Exit F4=Prompt F6=Save Mapping and Populate Files  
F8=Specify Files F9=Show Data F10=Display data F11=Populate Files  
F12=Cancel F16=Prompt All

---

q.v.. [Description of the Screens - Parser, XML Tag to Field Mapping](#)

We leave the screen by pressing the function key **F6(Save Mapping and Populate Files)** and enter the screen [I4XW001](#); there we can call an existing mapping or create a new mapping

## 5.4.8 Save Mapping And Populate Files

Here we can create a new mapping (e.g. PDEMO) or call an existing mapping. After pressing the **Enter Key** we save the newly created mapping and populate the physical files with the appropriate data and return to the [Parser Main Menu](#).

---

```
I4XW001
      Mapping Name List

Enter Mapping Name . . . . : _____

F3=Exit  F4=List Mapping  F12=Cancel
```

---

q.v.. [Description of the Screens - Generator, Mapping Name Screen](#)

## 5.5 Parser - Batch Processing

The Parser is started for the batch processing by using the command [PARSEXMLB](#) (**option 10** of the [Parser Main Menu](#)).

PARSEXMLB executes the first step (reading and analyzing the data) in batch processing mode. This option is used to parse XML documents with large amounts of data since it can be time intensive. Once the request is submitted, the user can continue working on other tasks.

The complete parameter description is described in the chapter: [The Command PARSEXMLB](#)

Once the parsing is completed, the user can work with this parsed output using the command [WRKPARSE](#) (**option 11** of the [Parser Main Menu](#)).

This process is exactly the same as the interactive command [PARSEXML](#).

The complete parameter description is described in the chapter: [The Command WRKPARSE](#)

## 6 The Commands - Overview

|                   |   |                            |
|-------------------|---|----------------------------|
| <b>Generator:</b> | - Create a XML Extraction Script (Extended Mode).           | <a href="#">GENXMLESC</a>  |
|                   | - Convert DB2 Data/ Generate XML.                           | <a href="#">EXTXMLDTA</a>  |
|                   | - Convert/ Generate XML (Fastpath).                         | <a href="#">GENXML</a>     |
| <b>Parser:</b>    | - Import XML-Data (interactive).                            | <a href="#">PARSEXML</a>   |
|                   | - Import XML-Data (Batch Processing).                       | <a href="#">PARSEXMLB</a>  |
| <b>Other</b>      |   |                            |
| <b>Commands:</b>  | - Define Parser Mapping after the Analyze is done in Batch. | <a href="#">WRKPARSE</a>   |
|                   | - Define larger Structures.                                 | <a href="#">WRKHRY</a>     |
|                   | - Evaluate XML file.  | <a href="#">EVALXML</a>    |
|                   | - Process multiple Input files.                             | <a href="#">MULTIPARSE</a> |
|                   | - Copy Parser-Mappings.                                     | <a href="#">CPYPRSMAP</a>  |
|                   | - Import Parser-Mappings.                                   | <a href="#">RSTPRSMAP</a>  |
|                   | - Maintain Generator-Mappings.                              | <a href="#">WRKGENMAP</a>  |
|                   | - Copy several XML-parts together                           | <a href="#">MERGEXML</a>   |

## 6.1 GENXMLESC

With the command GENXMLESC a XML Extraction Script is generated, which contains all information that is needed to extract data and create XML output with the command [EXTXMLDTA](#).

The command includes the following **parameters**:

### SCRIPT

*XML Script Name* Name of the script incl. path in exact IFS-Names, i.e. /home/xml/sample.esc

### INPUTTYPE

*Input Object Type* Defines the type of input file.

#### Valid entries:

\**QRYDFN* Input is an existing Query Definition.

\**FILE* Input is a single file.

\**MAPQRY* Input is a special, with i4XML created Query-Mapping file.

### QRYNAME

*Query Name*  
*/Library*

Name of the Query.  
Only valid for INPUTTYPE(\*QRYDFN)

### FILENAME

*File Name*  
*/Library*

Name of the input data.  
Only valid for INPUTTYPE(\*FILE)

### MAPNAME

*MAPQRY Name*

Name of the Mapping.  
Only valid for INPUTTYPE(\*MAPQRY)

#### Valid entries:

\**SELECT* A new mapping is generated with some dialogues. (Special value).

**HIRDEF***XML Hierarchy*

*Definition Name* Name of a previously defined hierarchy (with command [WRKHRY](#)).

**TAGNAM***XML Tag*

*Mapping Name* Name of a previously saved mapping.

Valid entries:

*\*CREATE* A dialogue is started to create a new Mapping.

*\*DEFAULT* A standard script will be generated without any dialogues.

*\*SELECT* A new mapping is generated with some dialogues (special value)

**RUNOPT***Replace*

Controls whether an existing script will be overwritten.

Valid entries: *\*YES* or *\*NO*

**MAPOPT***Mapping Load**Error Action*

This parameter defines what should happen in case of mapping loading errors.

Valid entries:

*\*BREAK* Processing will be cancelled in case of errors.

*\*IGNORE* Errors will be ignored.

**TAGSOURCE**

*Source  
of Tag names*

When a new mapping is created, i4XML determines the XML tag names either from the text of the file field description or from the field names itself.

Valid entries:

*\*TEXT* XML tag names are determined from the text of the file field description,  
i.e. "customer name" results in  
"<customer\_name>"

*\*NAME* XML tag names are determined from the field names itself,  
i.e. "CUSTNM" results in "<CUSTNM>"

## 6.2 EXTXMLDTA

With the command EXTXMLDTA the XML generation and conversion is executed based on a predefined extraction script.

The command includes the following **parameters**:

### SCRIPT

*Path of 'XML*

*Extractor Script*

Name of the script (generated with GENXMLESC) incl. path in exact IFS-Names, i.e. /home/xml/sample.esc

### PATH

*Path of XML*

Name of the XML file to be generated incl. path in exact IFS-Names, i.e. /home/xml/sampleoutput.xml

### GENDTD

*Generate DTD*

Controls whether a DTD (XML file description) is to be generated, i.e. /home/xml/sampleoutput.xml.

Valid entries:

*\*INT* DTD is generated internally, meaning within the newly generated XML file.

*\*EXT* DTD is generated externally, in a separate file (to be entered at parameter DTDPATH).

*\*WEB* A reference to an existing URL is generated into the XML file. At parameter DTDPATH a complete URL can be assigned.

*\*NO* No DTD.

### FTP

*Transfer by FTP*

Controls whether the generated file should be transferred to an Server by FTP

Valid entries: *\*YES* or *\*NO*.

**REPLACE***Replace File*

Controls whether an existing file will be overwritten.

Valid entries: \*YES or \*NO.**DTDPATH***Path Of DTD*Name of the DTD file to be generated incl. path in exact IFS-Names,  
i.e. /home/xml/sampledesc.dtd**FTPTOF***To FTP Path*

Path where the file has to be FTP-ed to.

**RMTSYS***Remote System*

Name of the FTP-Server.

Valid entries:*\*INTERNETADR* Instead of a server name an IP-address is to be used. (Special value)**INTERNETADR***IP address*

IP Address

**USRID***User*

FTP User

**PASWRD***Password*

FTP Password

**XSLPATH***Path Of XSL*

This parameter can be used to reference a XSL file.

**SLTRCD***Additional**SELECT Criteria*

Additional WHERE condition, which must be syntactically correct SQL.

**MASTTAG***MASTER Tag*

Tag name, which leads the XML structure.

Valid entries:*\*DEFAULT* A standard name is inserted.*\*NONE* No name is inserted. The tag is determined from the structure.**RECLVLTAG***Record Level Tag*

Tag name that leads each record.

Valid entries:*\*DEFAULT* A standard name is inserted.*\*NONE* No name is inserted. The tag is determined from the structure.**USESCH***Use**Existing Schema*

For XML files which references the special Schema files, i.e. „x-schema:OpenShipments.xdr“

Valid entries:*\*NONE* No Scheme to be used.*\*NOHEAD* No XML-Head is generated. Use this to generate XML parts which can later be combined.  
See command [MERGEXML](#) for more informations on this.**XMLENC***XML Encoding*

Encoding entries.

Valid entries:*\*DEFAULT* A standard entry is inserted:  
encoding="ISO-8859-1"

**DECFMT***Decimal Format*

Controls the „decimal indicator“.

Valid entries:

- . Decimal indicator is a period.
- , Decimal indicator is a comma.

**I4CLIENT***Start i4Client*

Starts i4Client.

This parameter controls the communication with the free Windows Client "i4Client" . It can used for starting a download in the temporary windows directory after the generation of the PC-file and opening the corresponding program.

Valid entries:

- \*LOCAL*           The XML file is presented at the local PC.
- \*LOCALSUI*       For internal use only.
- \*NONE*            No download.

**ONE2ONE***One Input Record  
to one stuct*

Controls whether each input record should result in one XML structure, regardless of whether there are any structures defined.

Valid entries: *\*YES* or *\*NO*.**UCCSID***Use CCSID*

Special use of CCSID or Codepage.

Valid entries:

- \*DEFAULT*       The current CCSID is used.

**GENTYPE***Generator Type*

Type of the generator.

Valid entries:*\*DOM* Standard method with IBM API.*\*OWN* Own method (about 50x faster).

## 6.3 GENXML

With the command GENXML a XML file can be generated from a physical file without a dialogue or a predefined creation of an extraction script.

The command includes the following **parameters**:

### SCRIPT

*XML Script Name* Name of the XML file to be generated incl. path in exact IFS-Names,  
i.e. /home/xml/sampleoutput.xml

### OBJTYP

*Data Object Type* Defines the type of file.

#### Valid entries:

\**QRYDFN* Input is an existing Query Definition.

\**FILE* Input is a single file.

\**MAPQRY* Input is a special, with i4XML created Query-Mapping file.

### OBJNAM

*Query Name/Library* Name of the Query.  
Only valid for OBJTYP(\*QRYDFN)

### OBJNAM1

*File Name/Library* Name of the Input file.  
Only valid for OBJTYP(\*FILE)

### OBJNAM2

*MAPQRY Name* Name of the mapping.  
Only valid for OBJTYP(\*MAPQRY)

### TAGNAM

*XML Tag*  
*Mapping Name* Name of the Tag-Mapping

#### Valid entries:

\**DEFAULT* A standard script will be generated.

**GENDTD***Generate DTD*

Controls whether a DTD (XML file description) is to be generated.

Valid entries:

- \**INT* DTD is generated internally, meaning within the newly generated XML file.
- \**EXT* DTD is generated externally, in a separate file (to be entered at parameter DTDPATH).
- \**NO* No DTD.

**FTP***Transfer by FTP*

Controls, whether the generated file should be transferred to a server by FTP.

Valid entries: \**YES* or \**NO*

**FTPTOF***To FTP Path*

Path where the file is to be FTP-ed to.

**RMTSYS***Remote System*

Name of the FTP-Server.

Valid entries:

- \**INTNETADR* Instead of a server name an IP-address is to be used.

**INTNETADR***IP address*

IP address

**USRID***User*

FTP User

**PASWRD***Password*

FTP Password

**DTDP***Path Of DTD*

Name of the DTD file to be generated incl. path in exact IFS-Names,  
i.e. /home/xml/sampledesc.dtd

**MAPOPT**

*Mapping Load  
Error Action*

Controls what should happen when an error occurs during the load process of the mapping.

Valid entries:

\**BREAK* Processing will be cancelled in case of errors.

\**IGNORE* Errors will be ignored.

**SLTRCD**

*Additional  
SELECT Criteria*

Additional WHERE condition which must be syntactically correct SQL.

**MASTTAG**

*MASTER Tag*

Tag name that leads the XML structure.

Valid entries:

\**DEFAULT* A standard name is inserted.

\**NONE* No name is inserted. The tag is determined from the structure.

**RECLVLTAG**

*Record Level Tag*

Tag name that leads each record.

Valid entries:

\**DEFAULT* A standard name is inserted

\**NONE* No name is inserted. The tag is determined from the structure.

**USESCH**

*Use  
Existing Schema*

For XML files. Refers to special Schema files, i.e. „x-schema:OpenShipments.xdr“

**XMLENC***XML Encoding*

Encoding entries.

Valid entries:

*\*DEFAULT* A standard entry is inserted:  
encoding="ISO-8859-1"

**DECfmt***Decimal Format*

Controls the "decimal indicator".

Valid entries:

- . Decimal indicator is a period.
- , Decimal indicator is comma.

**I4CLIENT***Start i4Client*

Starts i4Client. This parameter controls the communication with the free Windows Client "i4Client". It can be used for starting a download in the temporary windows directory after the generation of the PC-file and opening the corresponding program.

Valid entries:

- \*LOCAL* The XML file is presented at the local PC.
- \*LOCALSUI* For internal use only.
- \*NONE* No download.

**RUNOPT***Replace*

Controls whether an existing script is to be overwritten.

Valid entries: *\*YES* or *\*NO***ONE2ONE***One Input Record  
to one structure*

Controls whether each input record should result in one XML structure, regardless if there are any structures defined.

Valid entries: *\*YES* or *\*NO***UCCSID**

*Use CCSID*

Special use of CCSID or Codepage.

Valid entries:

\**DEFAULT* The current CCSID is used.

## **TAGSOURCE**

*Source of*

*Tag names*

When a new mapping is created, i4XML determines the XML tag names either from the text of the file field description or from the field names itself.

Valid entries:

\**TEXT* XML tag names are determined from the text of the file field description,  
i.e. "customer name" results in  
"<customer\_name>"

\**NAME* XML tag names are determined from the field names itself,  
i.e. "CUSTNM" results in "<CUSTNM>"

## 6.4 PARSEXML

With the command PARSEXML the import of a XML file to adequate DB2 tables (and accordingly physical files) is carried out.

The command includes the following **parameters**:

### SCRIPT

*XML Document Name* Name of the XML file to be parsed incl. path in exact IFS-Names,  
i.e. /home/xml/sampleinput.xml

### VALXML

*Validate*

*XML against DTD* To validate XML against an existing DTD before parsing starts.

Valid entries: \*YES or \*NO

### MAPNAM

*XML Tag*

*Mapping Name*

Name of a previous saved mapping. If PARSEXML is executed in Batch mode, it is mandatory to enter an existing mapping.

Valid entries:

\*SELECT A new mapping can be defined with dialogues.

\*DEFAULT The conversion process is started without additional dialogues. Data is put directly into the file referred to by the parameter FILNAM (special value).

### SPLTMRG

*Consider Merge*

*/Split Mapping*

When a mapping is entered with this parameter, it can control whether the Merge/Split-Processing should be executed or not.

Valid entries: \*YES or \*NO

**ACTOPT***Ignore Error*

Determines if errors should be ignored.

Valid entries: \*YES or \*NO**SINGFIL***Copy**data into One file*

This parameter gives the opportunity to store a complete structure to one file. (No additional dialogues are needed for this)

Valid entries: \*YES or \*NO**CVTATRB***Convert Attributes  
to Element*

Controls whether XML attributes are handled the same way as elements are handled. That means attributes can be transferred to physical file fields as well.

Valid entries: \*YES or \*NO**DECFMT***Decimal Format*

Controls the "decimal indicator".

Valid entries:

- . Decimal indicator is a period.
- , Decimal indicator is a comma.

**FILNAM***Enter new File Name  
/Library*

File to take the complete structure in the case of SINGFIL(\*YES) and MAPNAM(\*DEFAULT)

**CVTSPEC**

*Convert special  
Characters:*

Conversion for special (German) characters, which are not correct converted although CCSID is corrected.

Valid entries:

\*YES Characters are converted.

\*NO Characters are not converted.

**BOOST**

*Boost processing*

Special Parser call, which is optimized for larger files with no complex structures.

Valid entries: \*YES or \*NO

**UCCSID**

*Use CCSID*

If a special CCSID is to be used, it can be referred here.

Valid entries:

\*DEFAULT The current CCSID is used.

**RMVDTYPE**

*Remove DOCTYPE  
entry*

If a file contains a DOCTYPE-instruction, processing can be run into problems. Therefore, *i4XML* allows a manipulation of the input file.

Valid entries:

\*YES DOCTYPE instruction will be removed from the original file, no way back!

\*NO DOCTYPE stays in file. Processing is probably not possible.

**RPLUTF8**

*Replace UTF8  
to EBCDIC*

Special characters handling (UTF-8)

Valid entries:

\*YES Characters are converted.

\*NO Characters are not converted.

**UTF8ISO**

*Replace UTF8  
to ISO*

UTF-8 coded Input files are converted to ISO before the Parser starts. **Attention!!!** File will be replaced.

Valid entries: \*YES or \*NO

**ENCODING**

*Use Encoding*

With this encoding an incorrect encoding (i.e. "UFT-16") can be replaced with an encoding that can be processed (i.e. "ISO-8859-1").

Valid entries:

\*DEFAULT The encoding is not changed.

**ID2ELEM**

*Convert ID-Attrib.  
to Element*

Special Pre-Processing which converts <Id>-Tags into Attributes, so that they can process in a proper way.

Valid entries: \*YES or \*NO

## 6.5 PARSEXMLB

With the command PARSEXMLB the import of a XML file to adequate DB2 tables (and accordingly physical files) is carried out. This command is used in combination with [WRKPARSE](#) for reading in large files and if no mapping is existing. In this manner neither screen-session is blocked.

The command includes the following **parameters**:

### SCRIPT

#### XML

*Document Name* Name of the XML file to be parsed incl. path in exact IFS-Names,  
i.e. /home/xml/sampleinput.xml

### VALXML

#### Validate XML

##### *against DTD*

To validate XML against an existing DTD before parsing starts.

Valid entries: \*YES or \*NO

### CVTATRB

#### Convert Attributes

##### *to Element*

Controls whether XML attributes are handled the same way as elements are handled. That means attributes can be transferred to physical file fields as well.

Valid entries: \*YES or \*NO

### DECfmt

#### Decimal Format

Controls the "decimal indicator"

Valid entries:

- . Decimal indicator is a period.
- , Decimal indicator is a comma.

**JOBQ***Job-Queue*

Name of the JOBQ which the job will be submitted to.

Valid entries:

\**JOBQ* The JOBQ will be determined according to the job description of the current job.

\**INTER* Interactive processing (special value).

**UCCSID***Use CCSID*

Special use of CCSID or Codepage.

Valid entries:

\**DEFAULT* The current CCSID is used.

**ENCODING***Use Encoding*

With this encoding an incorrect encoding (i.e. "UFT-16") with an encoding that can be processed (i.e. "ISO-8859-1").

Valid entries:

\**DEFAULT* The encoding is not changed.

**CVTSPEC***Convert special Characters*

Conversion for special (German) characters, which are not correct converted although CCSID is corrected.

Valid entries:

\**YES* Characters are converted.

\**NO* Characters are not converted.

**RMVDTYPE***Remove**DOCTYPE entry*

If a file contains a DOCTYPE-instruction, processing can be run into problems. Therefore, *i4XML* allows a manipulation of the input file.

Valid entries:

\*YES DOCTYPE instruction will be removed from the original file, no way back!

\*NO DOCTYPE stays in file. Processing is probably not possible.

**RPLUTF8***Replace UTF8**to EBCDIC*

Special characters handling (UTF-8)

Valid entries:

\*YES Characters are converted.

\*NO Characters are not converted.

**UTF8ISO***Replace UTF8**to ISO*

UTF-8 coded Input files are converted to ISO before the Parser starts. **Attention!!!** File will be replaced.

Valid entries: \*YES or \*NO**ID2ELEM***Convert ID-Attrib.**to Element*

Special Pre-Processing which converts <Id>-Tags into Attributes, so that they can be processed in a proper way.

Valid entries: \*YES or \*NO

## 6.6 WRKPARSE

With the command WRKPARSE it is possible after the conclusion of the file analysis to continue to work with the command [PARSEXMLB](#) and define structures etc. The following is exactly equivalent to the interactive command [PARSEXML](#).

The command includes the following **parameters**:

### SCRIPT

*XML*

*Document Name* Name of the XML file to be parsed incl. path in exact IFS-Names,  
i.e. /home/xml/sampleinput.xml

### MAPNAM

*XML Tag*

*Mapping Name* Name of a previous saved mapping. If PARSEXML is executed in Batch mode, it is mandatory to enter an existing mapping.

Valid entries:

- \**CREATE* A new mapping can be defined with dialogues.
- \**DEFAULT* The conversion process is started without additional dialogues. Data is put directly into the file referred by the parameter FILNAM (special value).
- \**SELECT* Shows a list of existing Mappings.

### SPLTMRG

*Consider Merge*  
*/Split Mapping*

When a mapping is entered with this parameter, it can control whether the Merge/Split-processing should be executed or not.

Valid entries: \*YES or \*NO

### ACTOPT

*Ignore Error*

Determines if errors should be ignored.

Valid entries: \*YES or \*NO

**SINGFIL**

*Copy data  
into One file*

This parameter gives the opportunity to store a complete structure to one file. (No additional dialogues are needed for this)

Valid entries: \*YES or \*NO

**FILNAM**

*Enter new  
File Name/Library*

File to take the complete structure in the case of SINGFIL(\*YES)

## 6.7 WRKHRY

The command WRKHRY is used for the definition of complex structures. The result of this definition can then be used in the command [GENXMLESC](#) by the mapping definition.

The command includes the following **parameters**:

### HRYPDEF

*Hierarchy Definition* A new or existing Hierarchy name.

#### Valid entries:

\*ANY A dialogue appears with all existing mappings.

From the [Generator Main Menu](#) you can get to here in the screen "[Hierarchy List](#)" with **option 50**

---

```

I4XS022                Hierarchy List

2=Edit 3=Copy 4=Delete
Select                Hierarchy Name

                        CUSTOMER

                                                                A
                                                                *
                                                                *
                                                                *
                                                                *
                                                                *
                                                                *
                                                                *
                                                                V

F5=Refresh F6=Create F12=Cancel

```

---

q.v. [Description of the Screens - Generator, Hierarchy List](#)

## 6.8 EVALXML

The command EVALXML is used for the textual verifying of a XML file.

The command includes the following **parameters**:

### SCRIPT

#### XML

*Document Name*      Name of the XML file to be checked incl. path in exact IFS-Names,  
i.e. /home/xml/sampleinput.xml

### VALXML

#### *Validate XML against DTD*

To validate XML against an existing DTD before parsing starts.

Valid entries: \*YES or \*NO

## 6.9 MULTIPARSE

The command MULTIPARSE is able to work off a whole group of input files. Internally the command [PARSEXML](#) is actually being used.

The command includes the following **parameters**:

### **DIRNAME**

*Directory to scan*      Name of the path which contains the input files, i.e. /home/xml

### **EXT2SCAN**

*Extension to scan*      Extension to be scanned i.e. .xml

### **ACTION**

*Processing Action*      Controls what should happen with the input file after processing.

#### Valid entries:

\**MOVE*      File is moved after processing, accordingly to DIR2MOVE and MOVACT.

\**CHGORG*      File name or extension is changed after processing, accordingly to the parameter EXT2CHG, STAMPTYPE and STAMPPLACE.

\**DELETE*      File is deleted after processing.

### **DIR2MOVE**

*Directory to move files to*      Directory where the files should be moved to.  
Only valid when ACTION(\*MOVE)

**MOVACT***Action after Move*

Controls, what should happen with the file after the Move operation.

Valid entries:

- \**NONE* File is not manipulated further after the move.
- \**CHGEXT* Extension of the file is manipulated according to EXT2CHG.
- \**STAMP* File gets a time stamp according to STAMPTYPE.
- \**STAMPEXT* Extension of the file is manipulated according to STAMPTYPE

**EXT2CHG***Extension**to change to*

New extension. Only valid when ACTION(\*CHGORG) and accordingly MOVACT(\*CHGEXT)

**STAMPTYPE***Type of Stamp*

Type of stamp.

Valid entries:

- \**DATTIM* Timestamp is both date and time.
- \**DATE* Timestamp is date only.

**STAMPPLACE***Where to stamp*

Controls, where the stamp is inserted.

Valid entries:

- \**BEGIN* At the beginning of the name.
- \**END* At the end of the name.
- \**EXT* At the extension.
- \**NONE* No manipulation.

**MAPNAM***XML Tag**Mapping Name*

Name of a previously saved mapping.

**DECfmt***Decimal Format*

Controls the "decimal indicator".

Valid entries:

- . Decimal indicator is a period.
- , Decimal indicator is comma.

**IDE2ELEM***Convert ID-Attrib.  
to Element*

Special Pre-Processing which converts &lt;Id&gt;-Tags into Attributes, so that they can process in a proper way.

Valid entries: \*YES or \*NO

## 6.10 CPYPRSMAP

The command CPYPRSMAP (Copy Parser Mapping) is provided to extract and copy Parser-Mappings from the original Environment to a transport library.

I.e. it can be used to export Mappings from a developer machine and then imported to the live machine with the command [RSTPRSMAP](#).

The command has the following **parameters**:

### **FROMLIB**

*original library* Original Library where the Mappings are stored.

### **MAPNAM**

*Mapping to copy* Name of the Mapping to be exported.

#### Valid entries:

\**SELECT* Shows a list of available Mappings.

### **LIBNAM**

*Library to hold mapping* Target library.

### **MBROPT**

*Copy option for target* Determines whether Data is to be added or replaced.

#### Valid entries:

\**ADD* Data will be added.

\**REPLACE* Existing Data will be replaced.

## 6.11 RSTPRSMAP

The command RSTPRSMAP (Restore Parser Mapping) enables the import of Parser Mapping Definitions from a previously saved library. The Mappings must be provided with the command [CPYPRSMAP](#).

With both the commands one can handle a test/ production scenario as well as a development/ runtime scenario.

The command has the following **parameters**:

### MAPNAM

*Mapping to restore* Mapping to be restored.

#### Valid entries:

\**ALL*: All existing Mappings will be restored.

\**SELECT* Select one Mapping from a list.

### LIBNAM

*Library  
with mapping*

Library which contains the Mapping files (to be provided with command [CPYPRSMAP](#)).

### CPYOPT

*Copy Option*

Determines whether Data is to be added or replaced.

#### Valid entries:

\**ADD* Data will be added.

\**REPLACE* Existing Data will be replaced.

## 6.12 WRKGENMAP

The command WRKGENMAP is provided to maintain the Generator Mapping Definitions.

The command has the following **parameters**:

### GENMAP

*Mapping Name* To assign an existing Mapping.

#### Valid entries:

\*ANY Shows a Dialogue from where a particular Mapping can be selected.

From the [Generator Main Menu](#) you can get to here in the screen "[Generator Mapping List](#)" with **option 51**.

---

```

I4XS023                               Generator Mapping List

3=Copy 4=Delete 7=Rename 8=Change Input File
Select                               Mapping Name

                                STRUCT                                A
                                TESTCZB                             *
                                UPSINPUT                            *
                                UPSINPUTCO                          *
                                UPSINPUTTP                           *
                                UPSINPUTUF                           *
                                                                *
                                                                :
                                                                :
                                                                V

F3=Exit F5=Refresh F12=Cancel

```

---

q.v. [Description of the Screens - Generator, Mapping List](#)

## 6.13 MERGEXML

This command enables you to combine multiple parts of an “XML“-skeleton together to one valid XML stream file. Up to 3 input files plus one existing file can be combined together.

The command has the following **parameters**:

**XMLA**

*XML Input File A*      First Part.

**XMLB**

*XML Input File B*      Second Part.

**XMLC**

*XML Input File C*      Third Part.

**XMLOUT**

*XML Output File*      Output File.

**REPLACE**

*Replace File*

Determines whether an existing file has to be replaced or should be used as a “Base”, new data is appended.

Valid entries:

\*YES      Existing File will be replaced.

\*NO      Existing File won't be replaced; an error will be indicated in case there is an existing file.

\*APPEND      New Data is added to the existing file.

**MRGTYPE**

*Merge Type*

Select the Processor.

Valid entries:

\*RPG      i4XML own Module (recommend).

\*QSH      Merge with SQH.

## 7 Description of the Screens - Generator

### 7.1 Overview

|  |  |
|--|--|
| <a href="#">Select Break Fields</a>      | Selection of the Breaks - Break Fields.  |
| <a href="#">Field to XML Tag Mapping</a> | Attribution of the XML tags..  |
| <a href="#">Mapping Name Screen</a>      | Create a new mapping and accordingly selection of an existing mapping from a list. |
| <a href="#">Mapping List</a>             | Shows a list of existing Mappings.   |
| <a href="#">Hierarchies Definition</a>   | Definition of complex structures, application by the mapping definition.           |
| <a href="#">Hierarchy List</a>           | Shows existing Hierachies. Enables you to edit, copy and delete Hierachies.        |
| <a href="#">Generator Mapping List</a>   | Maintain Generator Mappings.   |
| <a href="#">Change Input File Name</a>   | Change Input-Files of one Mapping.   |

## 7.2 Select Break Fields

Using this screen, it is defined at which fields a break in the XML output shall be carried out. For a multi-level break, the break level can be defined by the structure.

---

```

I4XS012                XML Generator                30.07.07
                        Select Break Fields          17:37:26

Extraction Script Name : /I4XML/FILE2XML.ESC

File      Field      Break Break      Parent
Name     Name Y/X/N/A Lvl Tag Name     Tag Name
YCUSTOMER CUSTOMER  N 000 YCU_CUSTOMER
YCUSTOMER NAME       N 000 YCU_NAME
YCUSTOMER CITY       N 000 YCU_CITY
YCUSTOMER STREET    N 000 YCU_STREET
YCUSTOMER STREET2   N 000 YCU_STREET2

                                           End
F3=Exit F6=Confirm F8=Show SQL F10=Load Mapping F11=Display Data F12=Cancel

```

---

### Field Description

|                       |   |
|-----------------------|---|
| <i>File Name</i>      | Input file  |
| <i>Field Name</i>     | Field names of the input file.  |
| <i>Break Y/X/N/A</i>  | Break yes or no.  |
| Y                     | A Break is performed only if the content of the field changes.  |
| X                     | A Break is performed for each record, regardless of content change.   |
| N                     | Field is not a Break-Field.   |
| A                     | This field is an additional field and works together with the other field assigned with "Y" (sample: Customer-Type together with Customerid). |
| <i>Break Lvl</i>      | The level of the break field.   |
| <i>Break Tag Name</i> | If a field is changing to a break field, it gets a break tag name,  |

i.e. the name of the substructure is fixed.

*Parent Tag Name* This is the name of the higher level structure, i.e. the break tag name of the directly higher level.

### Function Keys

*F4=Hierarchy* Select the Hierarchy Tag from a list.  
This is only available if a Hierarchy is entered when the process is started at command level.

*F6=Confirm* Validation of the settings made.

*F8=Show SQL* Display the underlying SQL command.

*F10=Load Mapping* Selection of an existing mapping from a list.

*F11=Display Data* Display data.

## 7.3 Field to XML Tag Mapping

Here the input fields are classified to the relative structure. The level and the "Hierarchy" can be changed. But both will be initialized by the order of the input file.

---

```

I4XS014                                XML Generator                30.07.07
                                         Field to XML Tag Mapping      17:48:35

Extraction Script Name : FILE2XML.ESC

File      Field      Type Len  Dec XML Tag Name                Tag Type Lvl
Hierarchy
YCUSTOMER CUSTOMER  A     7   Customer                    E    002
/YCU_CUSTOMER
YCUSTOMER NAME      A    30   Name                        E    002
/YCU_CUSTOMER
YCUSTOMER CITY      A    20   City                        E    002
/YCU_CUSTOMER
YCUSTOMER STREET    A    20   Street                      E    002
/YCU_CUSTOMER
YCUSTOMER STREET2   A    20   Street2                      E    002
/YCU_CUSTOMER

                                         End
F3=Exit F4=Hierarchy F8=Show SQL F9=Save Mapping and Generate Extraction Script
F11=Generate Extraction Script F12=Cancel F13=Repeat Level
F14=Save Changes and Edit Mapping

```

---

### Field Description

*File* File name

*Field* Field name

*Type* Field type (e.g. A=alphanumeric., N=numeric)

*Len* Field length

*Dec* Number of decimal places

*XML Tag Name* The XML tag name of the generated XML file.

*Tag Type* The tag type of the generated XML file.

E = element

A = attribute

I = Internal, useful for situations where Field is a Breaktag and has attributes.

XML-Sample: <City ZIP="4711">New York</City>

*Lvl* The level of the individual tags, i.e. the hierarchy within the generated XML file.

*Hierarchy* The level on which this field is attributive.

### Function Keys

*F4=Hierarchy* Select the Hierarchy Tag from a list.  
This is only available if a Hierarchy is entered when the process is started at command level.

*F8=Show SQL* Display the underlying SQL query.

*F9=Save Mapping  
and Generate  
Extraction Script* Save the generated settings as mapping and generate the so-called "Extraction Script".

*F11=Generate  
Extraction Script* Generate the Extraction Script without saving of the mapping.

*F13=Repeat Level* Repeats the Level from the Field before.

*F14=Save Changes  
and Edit  
Mapping* Saves the settings and go back to Level Definitions.

## 7.4 Mapping Name Screen

Enter a new Mapping name or select an existing one from a list with **F4**.

---

```
I4XW002      Mapping Name Screen
Entering Mapping Name: _____
F4=List Mapping  F12=Cancel
```

---

### Field Description

*Entering Mapping Name*      Specification of a mapping name, if a new mapping is created (max. 10 digits), i.e. FILE2.

### Function Keys

*F4=List Mapping*      An existing mapping can be selected from the mapping list.

## 7.5 Mapping List

Shows a list of existing Mappings.  
With **option 1** the desired Mapping is selected.

---

```
I4XV002      Mapping List

1=Select
Option          Mapping Name

                STRUCT          A
                TESTCZB         *
                UPSINPUT        *
                UPSINPUTCO      *
                UPSINPUTTP      *
                UPSINPUTUF      :
                                :
                                :
                                :
                                V

F12=CANCEL
```

---

## 7.6 Hierarchies Definition

The generation of a new Hierarchy.

---

```

I4XS020                                XML Generator                27.08.07
                                      Hierarchies Definition        11:15:37
Hierarchy Definition : CUSTOMER
Type options, press Enter.
  1=Insert    4=Delete

Opt   Order   Level   Break Tag
     10       1      Customer_Detail
     20       2      Id_n_other
     30       2      Name_n_other
     40       3      Streets

F3=Exit    F5=Reorder    F6=Save    F17=Reorganize    Weitere ...

```

---

### Options

*1=Insert*            Insert a row BEFORE the current row.

*4=Delete*            Delete the current row.

### Field Description

*Opt*                    Selection of options.

*Order*                 Sequence number, can also be reordered by the function key **F5**.

*Level*                 Level of the hierarchy, can also be reorganized by the function key **F17**.

*Break Tag*            Name of the hierarchy tag.

### Function Keys

*F5=Reorder*            The Sequence number can be reordered with this key.

*F6=Save*                Save the new generated hierarchy.

*F17=Reorganize*        Reorganization of the hierarchy level.

## 7.7 Hierarchy List

Show all existing Hierarchies. From here one can edit, copy and delete the Hierarchy.

---

```

I4XS022                Hierarchy List
2=Edit 3=Copy 4=Delete
Select                Hierarchy Name
                        CUSTOMER
                                                A
                                                *
                                                *
                                                *
                                                *
                                                *
                                                *
                                                V
F5=Refresh F6=Create F12=Cancel

```

---

### Options

- |                 |                     |
|-----------------|---------------------|
| <i>2=Edit</i>   | Edit a Hierarchy.   |
| <i>3=Copy</i>   | Copy a Hierarchy.   |
| <i>4=Delete</i> | Delete a Hierarchy. |

### Field Description

*Hierarchy Name* Name of the Hierarchy.

### Function Keys

- |                   |                        |
|-------------------|------------------------|
| <i>F5=Refresh</i> | Load Subfile.          |
| <i>F6=Create</i>  | Create a New Hierarchy |



## 7.9 Change Input File Name

With this screen it is able to change the input file of a Mapping.

**Attention!!!** Fieldnames of old and new files must match; otherwise errors will occur when using the mapping at Parser.

---

```

I4XS024                               Change Input File

Mapping Name: TESTCZB

File Name      Library Name
YCUSTOMER     I4XML

A
*
*
*
*
*
V

F3=Exit F5=Refresh F9=Save Mapping and Regenerate Extraction Script
F10=Save Mapping F12=Cancel

```

---

### Function Keys

|   |  |
|---|--|
| <i>F5=Refresh</i>   | Reload the Setting.  |
| <i>F9=Save Mapping<br/>and Regenerate<br/>Extraction Script</i> | Saves the changes and Regenerates the Extraction Script, so that changes will affect the Generator.<br>(q.v. command <a href="#">EXTXMLDTA</a> )       |
| <i>F10=Save Mapping</i>   | Saves the changes in the Mapping file. This has no direct effect to the Extractions script and so no effect to the command <a href="#">EXTXMLDTA</a> . |

## 8 Description of the Screens - Parser

### 8.1 Overview

|   |  |
|---|--|
| <a href="#">Confirm XML Levels</a>          | Overview of the XML level.                                       |
| <a href="#">Split Level Details</a>         | Split one Level into new Sublevel.                               |
| <a href="#">Next Level Details</a>          | Working on the individual level.                                 |
| <a href="#">Element Data</a>                | Displaying the data content of individual fields.                |
| <a href="#">Specify Output Files</a>        | Entering of data information.                                    |
| <a href="#">Confirm File Specifications</a> | Creation of the physical files.                                  |
| <a href="#">XML Tag to Field Mapping</a>    | Attribution of the XML tags.                                     |
| <a href="#">Select Fields</a>               | Selection screen for the manual attribution tag - physical file. |
| <a href="#">Mapping List</a>                | Shows a list of existing Mappings.                               |
| <a href="#">Parser Mapping List</a>         | Maintain the Parser Mappings.                                    |

## 8.2 Confirm XML Levels

i4XML has the purpose, in the case of each level at a time, to transfer the data into a physical file. Because files without fields are not possible, such levels will not be possible.

You can influence the partitioning, as levels will be combined (Merge Level) or will be partitioned (Split Level).

---

```

I4XS001                                XML Parser                                21.06.04
                                        Confirm XML Levels                            10:08:21

XML File Name : /HOME/GOERING/XML/MANUAL/PARSERDEMO.XML
Type options, press Enter.

2=Split Level    5=Next Level Details    6=Merge Level    7=Hierarchy

Opt  Parent                Level    Relation
     CUSTOMER              02      N
     ORDER                 03      N
     LINE                  04      N
     OFFER                 03      N
     LINE                  04      N

F2=Show XML    F3=Exit    F6=Confirm    F10=Load Mapping    F12=Cancel    End

```

---

### Options

- 2=Split Level** Partitioning of the fields belonging to one level into several sub levels.
- 5=Next Level Details** Display the fields, which belong to the corresponding level, with the option to define Key-Fields, i.e. for each parent tag it is possible that the individual levels can be worked on separately.
- 6=Merge Level** Combination of levels.
- 7=Hierarchy** Display of the hierarchy.

**Field Description**

|                                    |  |
|------------------------------------|--|
| <i>Parent</i>                      | Under a "Parent" the appropriate fields are located (elements and attributes).       |
| <i>Level</i>                       | Here the levels are listed which were found during the analysis.                     |
| <i>Relation with Parents (Y/N)</i> | Specification, if a relationship with the predecessor level needs to be constructed. |

**Function Keys**

|                         |  |
|-------------------------|--|
| <i>F2=Show XML</i>      | Shows the underlying XML description.  |
| <i>F6=Confirm</i>       | Confirmation of the input after the editing of the individual levels (option 5) and the attribution of the Key-Fields.       |
| <i>F10=Load Mapping</i> | Display the screen I4XV001 " <a href="#">Mapping List</a> "; here you can choose one mapping from several existing mappings. |

## 8.3 Split Level Details

Here the fields of an existing Level can split into a Sublevel. The particular fields need to be selected together with a "1". The first field determines the Name of the Sublevel.

Another feature is the property to set one or more of the fields as a Key by assigning the Join Field with a "Y". These fields stay in the original level AND are generated to the new sublevel.

---

```

I4XS002                XML Parser                27.09.07
                        Split Level Details        10:43:51

Parent :  CUSTOMER                Level :  02
Type options, press Enter.
1=Select (Multiple Selections allowed)
Opt  Elements                Join
      CUSTOMERNR              Field (Y/N)
      NAME

F3=Exit    F6=Confirm    F10=Show Data    F12=Cancel

End

```

---

### Options

*1=Select*                    The field is removed from the current level and placed to the new sublevel In case of the "Join Field", the field remains in the original level.

### Field Description

*Parent*                    Current Parent.

*Level*                    Level of the current Parent.

*Elements*                Fieldname.

*Join Field*              Y =    Field is Key.

                            N =    Normal Data field

**Function Keys***F6=Confirm*

Confirms the Data entries.

*F10=Show Data*

Shows the data of a particular field (cursor sensitive)

## 8.4 Next Level Details

In the screen "Next Level Details", the subordinate levels of the selected parent tag (in the screen "Confirm XML Levels") are shown. These levels can be processed, e.g. the assigning of the key-fields for the subordinate levels.

As a rule, you have to construct a context between the different levels; therefore the fields will be specified as "General Key". This means that, in all subordinate levels, the "General Key" will be generated in the file.

---

```

I4XS003                XML Parser                21.06.04
                        Next Level Details        10:12:02

Hierarchy : /#document/PACKAGE/CUSTOMER
Parent    : CUSTOMER                          Level : 02

Elements                General Key (Y/N)

CUSTOMERNR
NAME

F3=Exit      F6=Confirm      F10=Show Data      F12=Cancel      End

```

---

### Field Description

|                          |  |
|--------------------------|--|
| <i>Hierarchy</i>         | Hierarchy level  |
| <i>Parent</i>            | Higher-ranking field (Parent Tag)  |
| <i>Level</i>             | Level rank   |
| <i>Elements</i>          | All fields of a hierarchy level will be displayed.   |
| <i>General Key (Y/N)</i> | Here the key fields for the subordinate levels will be specified. This is often necessary for producing a relationship in the physical file later. General Keys will be automatically adopted in each subordinate level (and also in their possible sub levels). |

### Function Keys

|                      |   |
|----------------------|---|
| <i>F6=Confirm</i>    | Confirmation of the input.                    |
| <i>F10=Show Data</i> | Show the data content of an individual field. |

## 8.5 Element Data

This screen is calling from the screen "[Next Level Details](#)" by pressing the function key **F10=Show Data**.

In the screen "Element Data" the data content of a field of a hierarchy level is been listed.

---

```
I4XS005                                XML Parser                21.06.04
                                         Element Data              10:13:50

Hierarchy : /#document/PACKAGE/CUSTOMER
Parent    : CUSTOMER                Level : 2
Element   : NAME

Row  Value
  1   Best Company

F3=Exit      F12=Cancel                                End
```

---

### Field Description

*Hierarchy* Current level with Hierarchy.

*Parent* Higher-ranking field (Parent Tag)

*Element* Tag name

*Level* Level

*Row* Row

*Value* Data content

## 8.6 Specify Output Files

After the applying of the Key-Fields a DDS will be generated and the physical files will be created.

In the screen "Specify Output Files" we will enter the data information (here in red font), which are necessary to create new files and accordingly overwrite the files in the IFS of the AS/400. Now the allocation of the level to the physical files will occur. If a file does not yet exist, it can be generated here "on the fly". Consequently, the option "Create File Y" will apply.

---

```

I4XS007                                     2.08.07
                                           Specify Output Files 12:39:49

      XML File  /HOME/GOERING/XML/MANUAL/PARSERDEMO.XML

Lvl Parent Tag      Library  File Name  Create File  Copy Option  Replace (Y)
  2  CUSTOMER      QTEMP    CUSTOMER   Y            *ADD        N
  3  ORDER        QTEMP    ORDER      Y            *ADD        N
  4  LINE         QTEMP    ORDERLINE  Y            *ADD        N
  3  OFFER        QTEMP    OFFER      Y            *ADD        N
  4  LINE         QTEMP    OFFERLINE  Y            *ADD        N
                                           End
Press enter to continue
F2=Parent Hierarchy  F7=Element Details  F12=Cancel  F13=Repeat

```

---

### Field Description

*Lvl*            Level

*Parent Tag*    Tag name

*Library*        Library, where the physical file has been located.

*File Name*     Name of the physical file.

*Create File*    Y    The physical file will be newly created.

                  N    The physical file will not be newly created.

                  X    Besides Y and N there is a special value "X". By using this value, the file will be created if it is not in existence already. In contrast to "Y" no error message will be appear, if the file already exists.

*Copy Option*    \*ADD            Records are added, existing data is kept in the file.

                  \*REPLACE    Existing data is cleared before adding new records.

*Replace (Y)*    Y    File will be replaced/ overwritten.

N File won't be overwritten.

### Function Keys

*F2=Parent Hierarchy* The Hierarchy will be shown in the screen I4XD003, e.g.  
/document/PACKAGE/CUSTOMER/OFFER/LINE

*F7=Element Details* Display the screen "[Next Level Details](#)"

*F13=Repeat* Repeat the input of the current column in all underlying rows.

## 8.7 Confirm File Specifications

In this screen the DDS description of the files is being defined, and will be created by *i4XML*.

At this time the field type and the field length can still be changed.

---

| I4XS010  | XML Parser                  | 2.08.07    |                     |      |     |     |            |
|--|-----------------------------|------------|---------------------|------|-----|-----|------------|
|  | Confirm File Specifications | 12:48:50   |                     |      |     |     |            |
| XML Name /HOME/GOERING/XML/MANUAL/PARSERDEMO.XML |                             |            |                     |      |     |     |            |
| Level  | Parent Tag<br>Library       | File       | Parsed Tag<br>Field | Type | Len | Dec | Datfmt     |
| 2  | CUSTOMER                    |            | CUSTOMERNR          |      |     |     |            |
|  | QTEMP                       | CUSTOMER   | CUSTOMERNR          | N    | 007 | 0   |            |
| 2  | CUSTOMER                    |            | NAME                |      |     |     |            |
|  | QTEMP                       | CUSTOMER   | NAME                | C    | 012 | 0   |            |
| 3  | ORDER                       |            | CUSTOMERNR          |      |     |     |            |
|  | QTEMP                       | ORDER      | CUSTOMERNR          | N    | 007 | 0   |            |
| 3  | ORDER                       |            | ORDERNR             |      |     |     |            |
|  | QTEMP                       | ORDER      | ORDERNR             | N    | 004 | 0   |            |
| 3  | ORDER                       |            | VALUE               |      |     |     |            |
|  | QTEMP                       | ORDER      | VALUE               | N    | 008 | 2   |            |
| 4  | LINE                        |            | CUSTOMERNR          |      |     |     |            |
|  | QTEMP                       | ORDERLINE  | CUSTOMERNR          | N    | 007 | 0   |            |
| 4  | LINE                        |            | ORDERNR             |      |     |     |            |
|  | QTEMP                       | ORDERLINE  | ORDERNR             | N    | 004 | 0   |            |
|  |                             |            |                     |      |     |     | More ...   |
|  | F2=Parent Hierarchy         | F6=Confirm | F10=Display Data    |      |     |     | F12=Cancel |

---

### Function Keys

**F2=Parent Hierarchy** The hierarchy is displayed in the screen I4XD004 "XML Parser (Hierarchy)", e.g. /#document/PACKAGE/CUSTOMER/OFFER/LINE

**F6=Confirm** Confirmation of the values.

**F10=Display Data** Display the screen "[Element Data](#)".

## 8.8 XML Tag to Field Mapping

In the screen "XML Tag to Field Mapping" all fields, which are necessary for the creation of the physical files, are displayed. The context between the XML tags and the fields of the physical file is delineated. Each tag of the XML file will be displayed and to which database field it is assigned. The new generated tags will be displayed in red font.

If the file has been generated with i4XML, the fields are already correctly filled in.

Otherwise the field assignment has to be performed manually; for that a straightforward [selection screen](#) is available by pressing the function key **F4**. For tags, which are not to be adopted, please define as "\*OMIT"

---

```

I4XS006                XML Parser                2.08.07
                    XML Tag to Field Mapping    12:59:47

                    XML File /HOME/GOERING/XML/MANUAL/PARSERDEMO.XML

Level  Parent Tag      Library      File      Parsed Tag      DB Field      ConvTyp      Type Flag
      2    CUSTOMER      QTEMP       CUSTOMER    CUSTOMERNR      CUSTOMERNR    _____  N   Y
      2    CUSTOMER      QTEMP       CUSTOMER    NAME            NAME          _____  C   N
      3    ORDER        QTEMP       CUSTOMER    CUSTOMERNR      CUSTOMERNR    _____  N   N
      3    ORDER        QTEMP       ORDER      ORDERNR         ORDERNR       _____  N   Y
      3    ORDER        QTEMP       ORDER      VALUE           VALUE         _____  C   N
      4    LINE         QTEMP       ORDERLINE  CUSTOMERNR      CUSTOMERNR    _____  N   N

```

More ...

```

F2=Parent Hierarchy F3=Exit  F4=Prompt F6=Save Mapping and Populate Files
F8=Specify Files   F9=Show Data F10=Display data F11=Populate Files
F12=Cancel        F16=Prompt All

```

---

### Field Description

|                   |  |
|-------------------|--|
| <i>Level</i>      | Level  |
| <i>Parent Tag</i> | Parent tag   |
| <i>Parsed Tag</i> | Tag name of the XML file.                              |
| <i>Type</i>       | Data type N=numeric, C=character/alphanumeric          |
| <i>Flag</i>       | Assigns whether data integrity checks are done or not. |
| <i>Library</i>    | Library of the Target File.                            |
| <i>File</i>       | Target File  |

|                 |  |
|-----------------|--|
| <i>DB Field</i> | Field of Target File.                                  |
| <i>ConvTyp</i>  | Assigns a special Conversion i.e. Date Conversion.     |
| <i>InputFmt</i> | Input format (in relation to the conversion parameter) |

### Function Keys

|   |   |
|---|---|
| <i>F2=Parent Hierarchy</i>                | The hierarchy will be displayed in the screen I4XD002 "XML Parser (Hierarchy)",<br>e.g.<br>/#document/PACKAGE/CUSTOMER/OFFER/LINE |
| <i>F4=Prompt</i>                          | Display a selection screen " <a href="#">Select Fields</a> " for the manual assignment of tags to the fields of a physical file.  |
| <i>F6=Save Mapping and Populate Files</i> | Save the mapping and populate the files.  |
| <i>F8=Specify Files</i>                   | Display the screen " <a href="#">Specify Output Files</a> " - change the data information.  |
| <i>F9=Show Data</i>                       | Calling the screen I4XS009 "Subfile Representation Of XML File Data" - the data content of all records will be displayed          |
| <i>F10=Display Data</i>                   | Calling the screen I4XS005 " <a href="#">Element Data</a> " - the data content of a selected field is displayed.                  |
| <i>F11=Populate Files</i>                 | Populate the files (the mapping hasn't been created as new).  |
| <i>F16=Prompt All</i>                     | Enables the function key <b>F4</b> , but here automatically for all fields.   |

## 8.9 Select Fields

In this selection screen a manual assignment of the tags to the fields of the physical file can be performed.

---

```

I4XS008                Select Fields                21.06.05
                                                         10:22:25

Type options, press Enter.
1=Select

Tag Name   :   NAME                Type   :   C
Library Name:  QTEMP                Length  :   012
File Name  :  CUSTOMER              Dec    :

      Field      ----Field----
Opt Name      Type Len Dec Field Description
-  CUSTOMERNR  S   007 0  CUSTOMERNR
_   NAME      A   012 0   NAME

                                                         End

Press enter to continue
F3=Exit

```

---

### Field Description

*Tag Name*            Tag name of the XML file.

*Library Name*        Target library.

*File Name*            Target file.

*Field Name*          Fieldname of Target file.

*Field Description*    Field Description

## 8.10 Mapping List

Show a list of existing Mappings to be selected with **Option 1**.

---

```
I4XV001
  Mapping List

Select Mapping Name

      CZBTEST      A
      MEINL2       *
      ODMASS       *
      ODMASS2      *
      UPSOUTPUT    :
      UPSOUTPUTN   :
      UPSOUTWO     :
      VPR          :
      VPR2         :
      VPS_UNLOAD   V

1=Select F12=Canc
```

---



## 9 Other i4XML tasks

---

```

I4XMLO                                the iSeries XML Tool!

Select one of the following:

    ** special purpose    **
    1. Validate XML-File

    ** system    **
    10. Change current path
    11. Explore IFS
    15. Work with Query

    ** i4XML-Administration **
    50. Setup i4XML
    51. Enter Licence Key
    52. Display version info
    60. See sample sources / SDK

Selection or command
===>

    F3=Exit          F4=Prompt.      F9=Retrieve      F12=Cancel
    F13=Information  F16=AS/400-Main Menu
  
```

---

- |                                     |   |
|-------------------------------------|---|
| 1. <u>Validate XML-File</u>         | The command <a href="#">EVALXML</a> evaluates a XML-File. |
| 10. <i>Change current path</i>      | CD command OS/400.  |
| 11. <i>Explore IFS</i>              | WRKLNK command OS/400.                                    |
| 15. <i>Work with Query</i>          | WRKQRY command OS/400.                                    |
| 50. <i>Setup i4XML</i>              | Calls <i>i4XML-Setup</i> .                                |
| 51. <i>Enter Licence Key</i>        | To enter the license Key for <i>i4XML</i> .               |
| 52. <i>Display version info</i>     | Show the version number.                                  |
| 60. <i>See sample sources / SDK</i> | Entry to SDK / opens PDM to work with samples.            |

# Index

## - 1 -

1-Step-Mode (Fastpath) 18, 34

## - A -

ADDLIBLE 15  
 adopt the settings 9, 10  
 allocation of the level to the physical files 97  
 assigning of the key-fields 95

## - B -

backup file 13  
 BRAVO Reader 8, 11  
 break  
   - field 24  
   - in the XML output 80  
   - level 80  
   - tag 24, 30  
   multilevel - 80  
 Bugbusters 12  
 Button "Code Only" 15

## - C -

CALL I4XMLCODE 15  
 CALL I4XMLSETUP 10  
 change  
   - field length 99  
   - field type 99  
 CHGCURDIR 16  
 command  
   ADDLIBLE 15  
   CALL i4XMLCODE 15  
   CALL I4XMLSETUP 10  
   CHGCURDIR 16  
   CRTDIR 16  
   CRTSAVF 13  
   EVALXML 71  
   EXTXMLDTA 18, 20, 27, 33, 34, 51  
   GENXML 18, 34, 56

GENXMLESC 18, 20, 30, 34, 48  
 GENXMLSESC 21  
 GO I4XML 17  
 MULTIPARSE 72  
 PARSEXML 38, 46, 61  
 PARSEXMLB 46, 65  
 RNMOBJ 9, 10  
 RSTLIB 13  
 STRTCPSVR 9  
 WRKHRY 18, 20, 28, 70  
 WRKPARSE 46, 68

complex structures 18  
 context between the different levels 41, 95  
 create an Extraction Script 20  
 CRTDIR 16  
 CRTSAVF 13

## - D -

definition  
   - of complex structures 70  
   - of the DDS description 99  
 DOS command entry window 13  
 download  
   - file 9  
 downloading from the internet 8

## - E -

EVALXML 71  
 example  
   - complex structure 28, 30, 33  
   - Generator 21, 23, 24, 25, 28, 30, 33  
   - Parser 36, 37, 38, 39, 40, 41, 42, 43, 44, 45  
 Extended Mode (Standard) 18, 20  
 Extract XML Data 27  
 EXTXMLDTA 18, 20, 27, 33, 34, 51

## - F -

false license key 15  
 file  
   - information 97  
   backup - 13  
   download- 9  
   I4XML 13  
   I4XML.BRV 9

file  
 I4XML.SAV 9, 12, 13  
 I4XML.SAVF 13  
 zip- 9

free download 12

## FTP

- service 9
- session 13
- techniques 8
- variant 9

## - G -

General Key 95

generate DDS 42, 97

generate the Extraction Script 48

- interactive 21
- manually written 23

## Generation

- of a XML file 20, 21
- of the XML data 20, 21

## Generator

- 1-Step-Mode (Fastpath) 18, 34
- Extended Mode 18, 20

GENXML 18, 34, 56

GENXMLESC 18, 20, 21, 30, 34, 48

## - H -

hierarchy 18

- name 28
- change - 82
- define a - 20, 30
- level of the - 86

## - I -

## i4XML

- installation 9
- library 9, 15
- license 15
- program 1, 7, 8, 9, 12, 13, 15, 17
- release 8, 10

I4XML.BRV 9

I4XML.SAV 9, 12, 13

I4XML.SAVF 13

I4XMLOLD 9

import of a XML file to DB2 tables 61, 65

INSTALL AID 8, 12

## installation 7

- BRAVO Reader Version 11
- downloading from the internet 8
- INSTALL AID 12
- manually transferred by FTP 13
- options 8
- preparations 9
- program 9
- recommendation 8, 9, 12

iSeries-Backup-File 8

## - L -

## level

- transfer into a physical file 91
- change - 82
- combine - 91
- context between the different - 95
- individual - 39, 40
- lower - 39, 40, 41
- partition - 91
- process - 95
- subordinated - 95

## library

- I4XML 9, 10, 13, 15
- I4XMLOLD 9, 10
- QGPL 13
- search list 15

## license 7

- key 15
- program 1

list the data content of a field 96

## - M -

manual assignment of the tags 102

## mapping

- definition 70
- name 25, 30, 84
- selection 24
- create a - 25, 30, 84
- existing - 25, 34, 44, 45, 84
- new - 25, 44, 45, 84

Merge Level 91

model-number 15

multi-level break 80

MULTIPARSE 72

## - N -

new generated tags 43

## - P -

parameter description 21, 30, 33, 34, 38, 46

parent tag 39, 41, 95

PARSEXML 38, 46, 61

PARSEXMLB 46, 65

physical files 44

creation of - 42, 43, 97, 100

populate - 45

proceeding

- for structures with high complexity 20

- for structures with normal complexity 20

processor-number 15

program version 1

## - Q -

QGPL 13

## - R -

reading in large files 65

release 7

RNMOBJ 9, 10

RSTLIB 13

## - S -

screen Generator

Extract XML Data (EXTXMLDTA) 33

Generate XML Extraction Script (GENXMLESC)  
21, 30

I4XS012 - Select Break Fields 24, 80

I4XS014 - Field to XML Tag Mapping 25, 82

I4XS020 - Hierarchies Definition 28, 86

I4XW002 - Mapping Name Screen 25, 84

screen Parser

I4XS001 - Confirm XML Levels 39, 91

I4XS003 - Next Level Details 40, 41, 95

I4XS005 - Element Data 40, 96

I4XS006 - XML Tag to Field Mapping 44, 100

I4XS007 - Specify Output Files 42, 97

I4XS008 - Select Fields 102

I4XS010 - Confirm File Specifications 43, 99

I4XW001 - Mapping Name List 45

Parse XML Data-Interactive (PARSEXML) 38

sequence number 86

serial-number 15

setup program 10

Split Level 91

STRTCPSVR 9

structure

- with high complexity 20, 28, 30, 33

- with normal complexity 20

define a - 46, 68, 80

system information 15

## - T -

tag

- level 25

- name 25

- type 25

textual verifying of a XML file 71

Transfer PC to the AS/400 13

## - U -

Usage

- Generator 3

- Parser 3

## - W -

WinZip 9

work off a whole group of input files 72

WRKHRY 18, 20, 28, 70

WRKPARSE 46, 68

## - X -

XML

- file 25, 100

- generation 56

- generation and conversion 51

- tag 100

XML Data Analysis

---

XML Data Analysis

- Batch Processing 35, 46

Interactive - 35, 36

## - Z -

Zip file 9