How to Import an XML file into an IBM i Database using CoolSpools

The CoolSpools Database module has new functionality which simplifies the process of importing XML files into the IBM i (AS400, iSeries) database. This easy to use and intuitive process consists of three steps:

Generate the map

By using a sample of the XML file to be imported, the CoolSpools 'Build XML-DBF Map' (BLDXMLDBF) command generates the XML-to-database map structure.

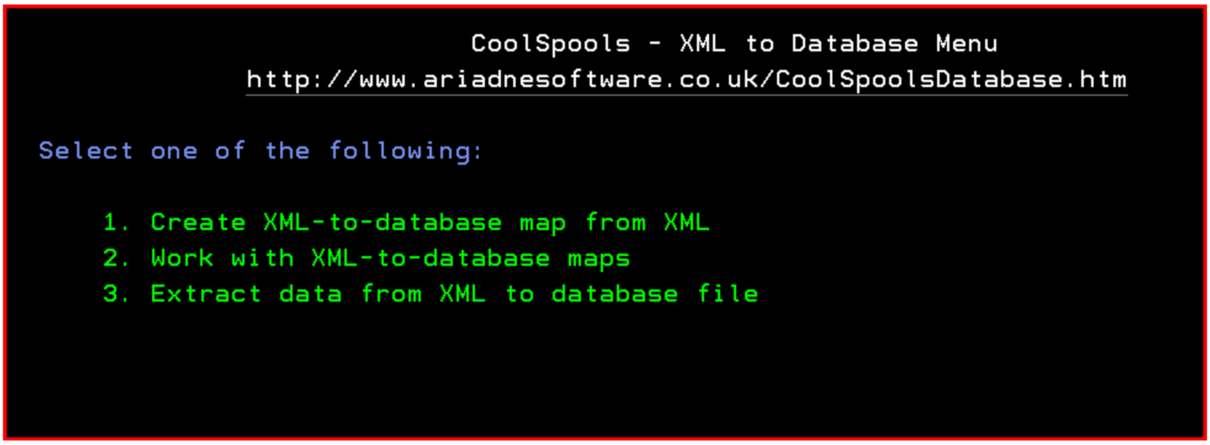
Map the XML elements and/or attributes to the IBM i database fields

The 'Work with XML-to-database maps' (WRKXMLDBF) command is used to edit the map created in step 1 and specify which database field should be populated by each XML element/attribute.

Import XML into the IBM i database

By specifying the IFS path to the XML document and the XML-to-database map name the Import XML to Database (IMPXMLDBF) command parses the XML and populates the IBM i database.

All the XML-to-database commands can be found on the CoolSpools XMLDBF menu, also available via Option 23 on the CoolSpools DATABASE menu.

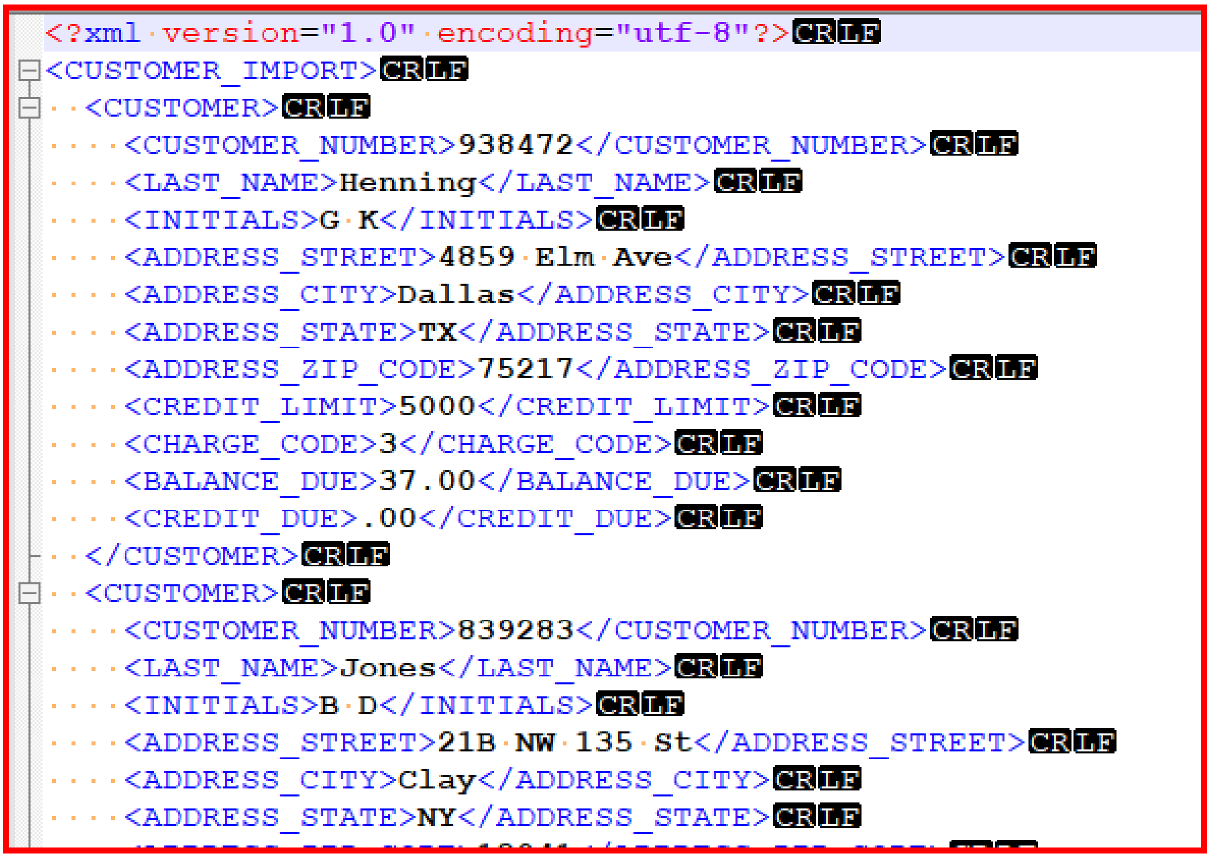


```
                           CoolSpools - XML to Database Menu
              http://www.ariadnesoftware.co.uk/CoolSpoolsDatabase.htm

 Select one of the following:

     1. Create XML-to-database map from XML
     2. Work with XML-to-database maps
     3. Extract data from XML to database file
```

Example

This example will use a simple XML document of Customer related data.

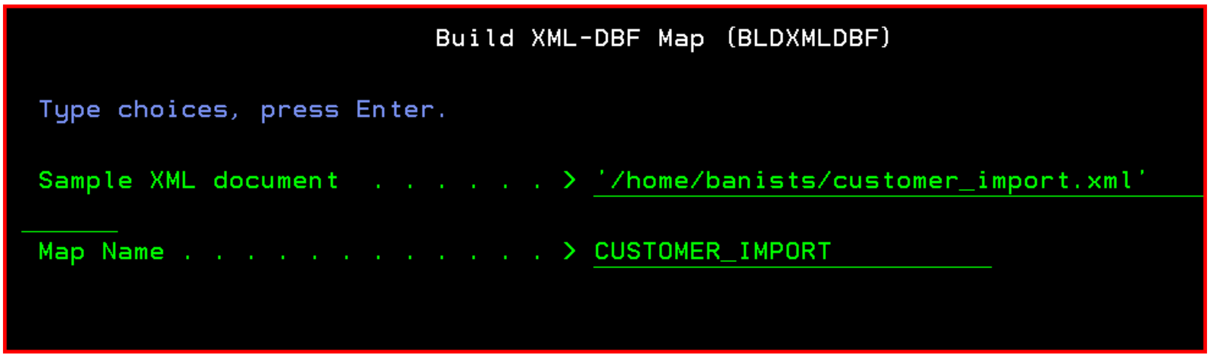
Here is the structure of the XML.



```
<?xml version="1.0" encoding="utf-8"?>
<CUSTOMER_IMPORT>
  <CUSTOMER>
    <CUSTOMER_NUMBER>938472</CUSTOMER_NUMBER>
    <LAST_NAME>Henning</LAST_NAME>
    <INITIALS>G K</INITIALS>
    <ADDRESS_STREET>4859 Elm Ave</ADDRESS_STREET>
    <ADDRESS_CITY>Dallas</ADDRESS_CITY>
    <ADDRESS_STATE>TX</ADDRESS_STATE>
    <ADDRESS_ZIP_CODE>75217</ADDRESS_ZIP_CODE>
    <CREDIT_LIMIT>5000</CREDIT_LIMIT>
    <CHARGE_CODE>3</CHARGE_CODE>
    <BALANCE_DUE>37.00</BALANCE_DUE>
    <CREDIT_DUE>.00</CREDIT_DUE>
  </CUSTOMER>
  <CUSTOMER>
    <CUSTOMER_NUMBER>839283</CUSTOMER_NUMBER>
    <LAST_NAME>Jones</LAST_NAME>
    <INITIALS>B D</INITIALS>
    <ADDRESS_STREET>21B NW 135 St</ADDRESS_STREET>
    <ADDRESS_CITY>Clay</ADDRESS_CITY>
    <ADDRESS_STATE>NY</ADDRESS_STATE>
```

Step 1 - Generate XML-to-database map from XML

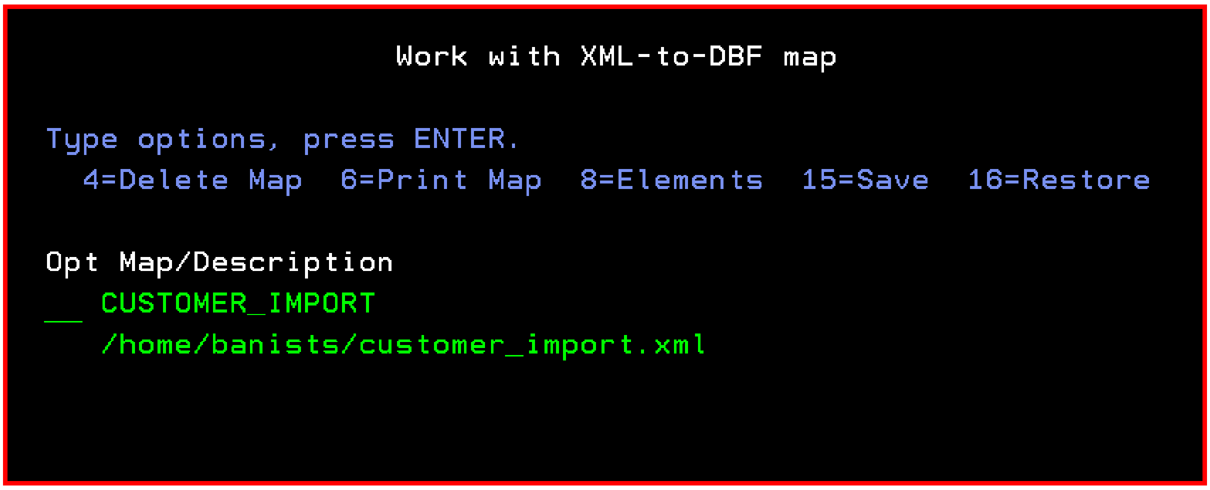
Menu option 1 will display the command prompt for BLDXMLDBF (Build XML-To-DBF map) which parses the specified sample example document and populates the CoolSpools map files to create a skeleton map based on its structure.



```
                         Build XML-DBF Map (BLDXMLDBF)

 Type choices, press Enter.

 Sample XML document . . . . . . > '/home/banists/customer_import.xml'

 Map Name . . . . . . . . . . . . > CUSTOMER_IMPORT
```

Step 2 - Map the XML elements and/or attributes to the IBM i database fields

Using Menu option 2 Work with XML-to-database maps, a list of maps is displayed.



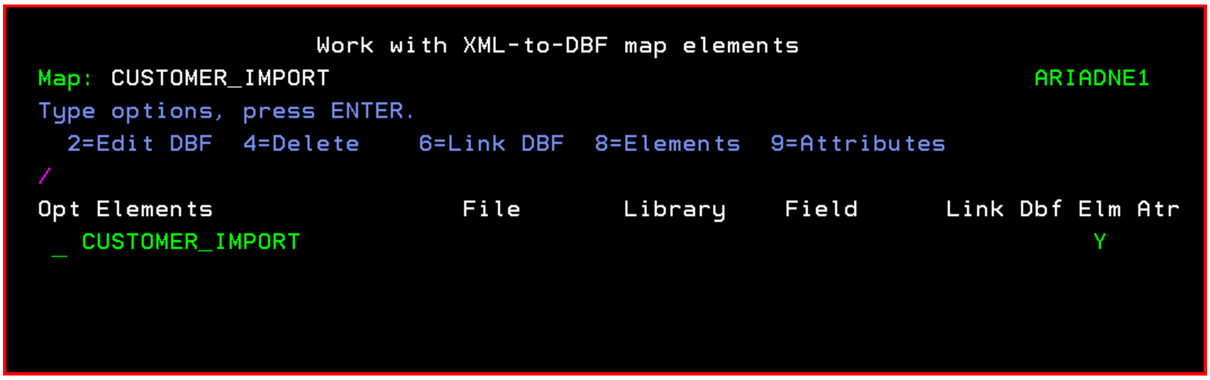
```
                    Work with XML-to-DBF map

 Type options, press ENTER.
   4=Delete Map  6=Print Map  8=Elements  15=Save  16=Restore

 Opt Map/Description
 __  CUSTOMER_IMPORT
     /home/banists/customer_import.xml
```

Options that can be input against the maps are:

4=Delete Map	Delete the selected map/s.
6=Print Map	Creates a spool file showing the XML structure and the
8=Elements	View and edit the structure of the XML map. The association of an XML element or attribute to a database field.
15=Save	Extracts the map and creates a save file.
16=Restore	The map previously saved recreated.

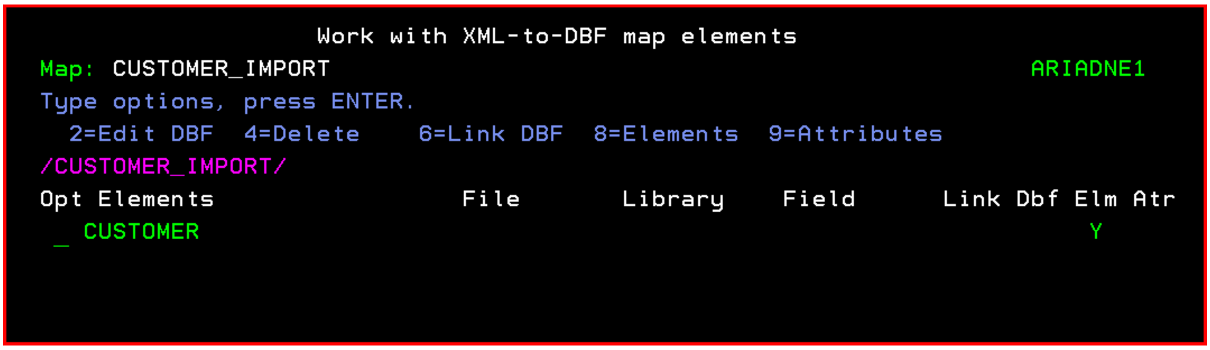
Review the map

To map the XML to the IBM i database fields select the map with option 8=Elements and the XML structure is displayed. For a full explanation of the screen please refer to the help text.



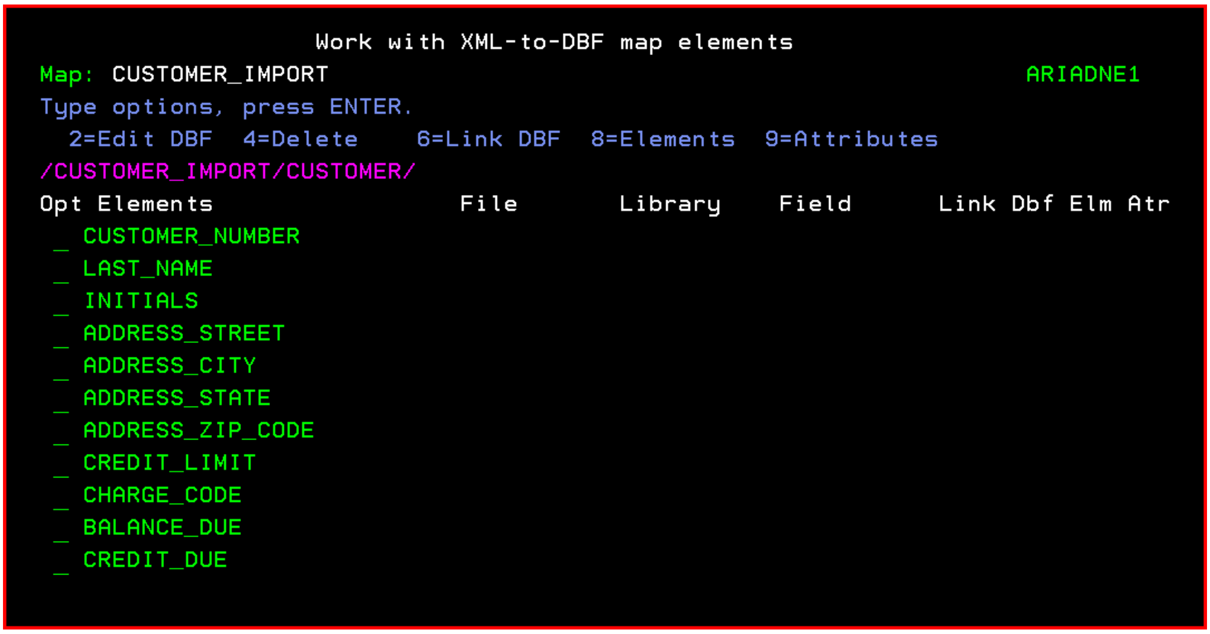
```
                   Work with XML-to-DBF map elements
Map: CUSTOMER_IMPORT                                          ARIADNE1
Type options, press ENTER.
  2=Edit DBF  4=Delete   6=Link DBF  8=Elements  9=Attributes
/
Opt Elements                  File       Library    Field       Link Dbf Elm Atr
 _ CUSTOMER_IMPORT                                                          Y
```

To display a lower level of the map, select the Element with option 8=Elements.



```
                      Work with XML-to-DBF map elements
Map: CUSTOMER_IMPORT                                                   ARIADNE1
Type options, press ENTER.
  2=Edit DBF   4=Delete    6=Link DBF   8=Elements   9=Attributes
/CUSTOMER_IMPORT/
Opt Elements                     File        Library     Field       Link Dbf Elm Atr
 _  CUSTOMER                                                                   Y
```

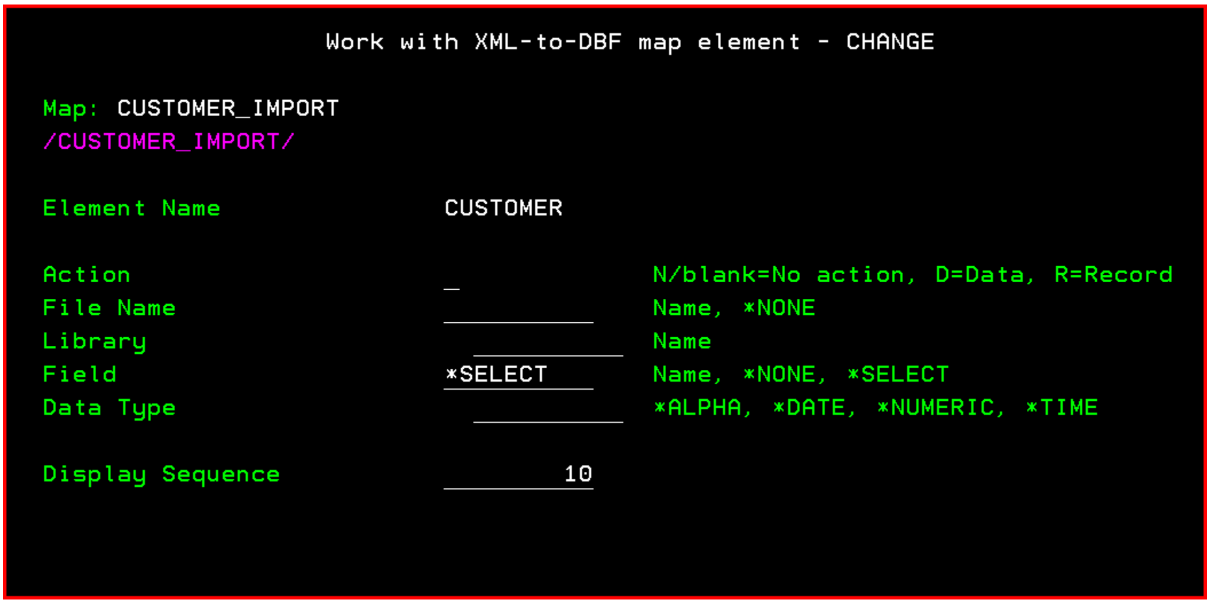
All the field level elements will be displayed.



```
                      Work with XML-to-DBF map elements
Map: CUSTOMER_IMPORT                                                   ARIADNE1
Type options, press ENTER.
  2=Edit DBF   4=Delete    6=Link DBF   8=Elements   9=Attributes
/CUSTOMER_IMPORT/CUSTOMER/
Opt Elements                     File        Library     Field       Link Dbf Elm Atr
 _  CUSTOMER_NUMBER
 _  LAST_NAME
 _  INITIALS
 _  ADDRESS_STREET
 _  ADDRESS_CITY
 _  ADDRESS_STATE
 _  ADDRESS_ZIP_CODE
 _  CREDIT_LIMIT
 _  CHARGE_CODE
 _  BALANCE_DUE
 _  CREDIT_DUE
```

Specify the database file

In this example, the data will be written to a database file called CUS_IMPORT.
To specify the database file, select an element that is at a higher level than the field level elements. In this case it will be the CUSTOMER element. Select it with option 2=Edit DBF.



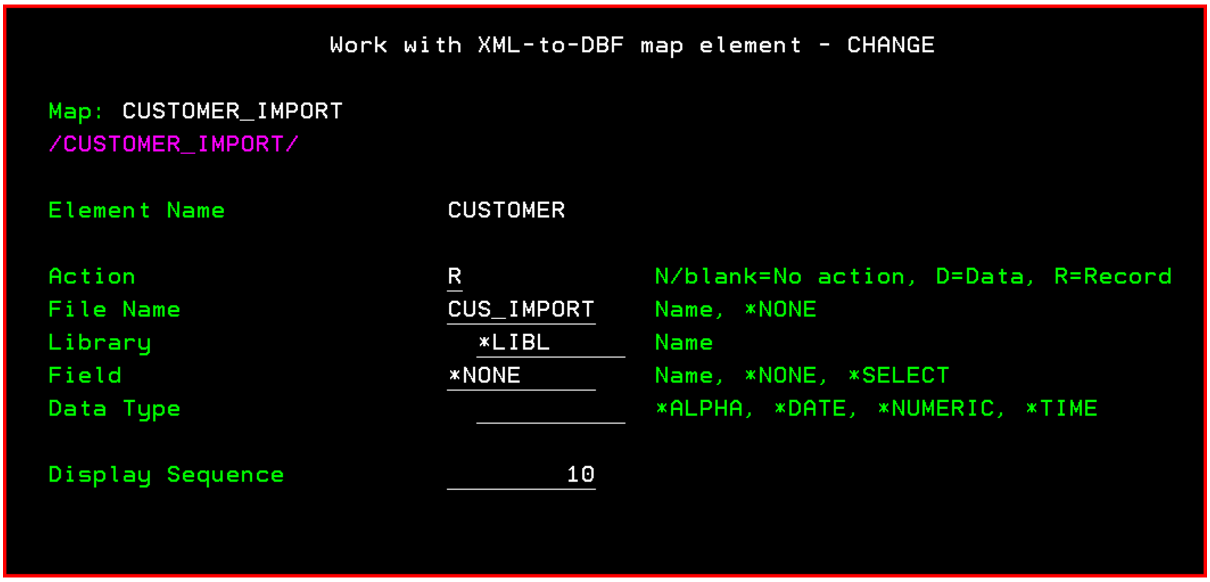
```
                  Work with XML-to-DBF map element - CHANGE

 Map:  CUSTOMER_IMPORT
 /CUSTOMER_IMPORT/

 Element Name                CUSTOMER

 Action                      _              N/blank=No action, D=Data, R=Record
 File Name                   __________     Name, *NONE
 Library                       __________   Name
 Field                       *SELECT___     Name, *NONE, *SELECT
 Data Type                     __________   *ALPHA, *DATE, *NUMERIC, *TIME

 Display Sequence                    10
```

Enter the following values:

Action: R Write Record. Populate the library/file specified. A record will be written to the data base file specified.

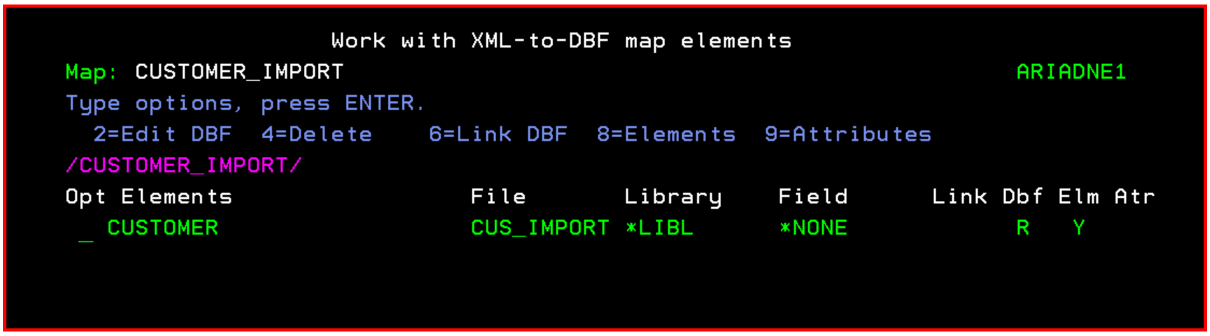
File Name and Library.

Field: *NONE



```
                  Work with XML-to-DBF map element - CHANGE

 Map:  CUSTOMER_IMPORT
 /CUSTOMER_IMPORT/

 Element Name                CUSTOMER

 Action                      R              N/blank=No action, D=Data, R=Record
 File Name                   CUS_IMPORT     Name, *NONE
 Library                       *LIBL_____   Name
 Field                       *NONE_____     Name, *NONE, *SELECT
 Data Type                     __________   *ALPHA, *DATE, *NUMERIC, *TIME

 Display Sequence                    10
```

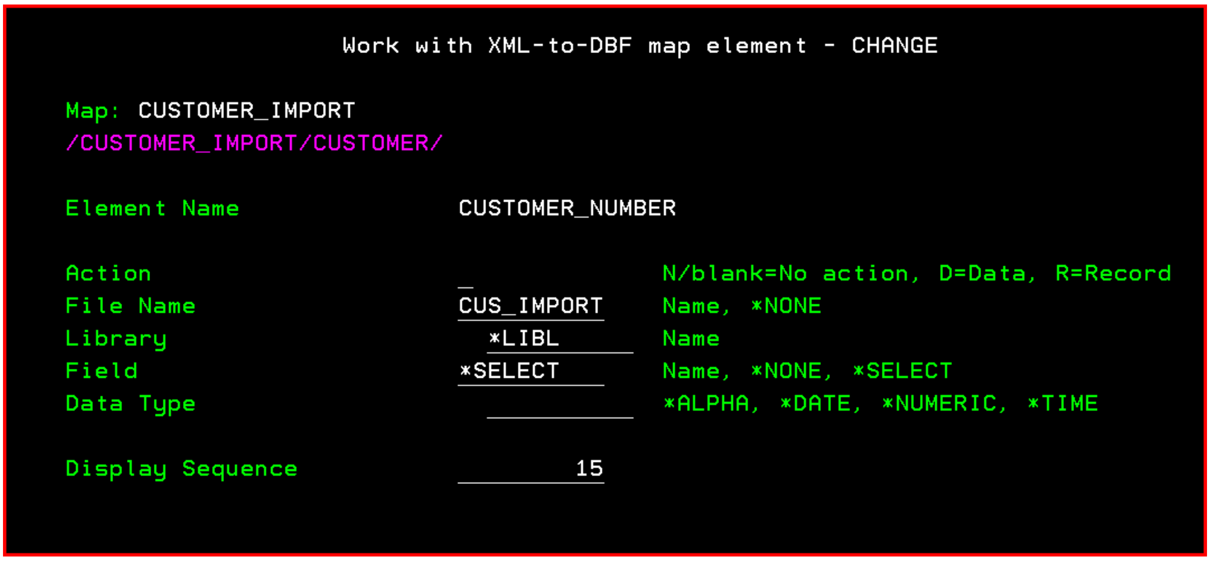
The element screen will be re-displayed



```
                   Work with XML-to-DBF map elements
Map: CUSTOMER_IMPORT                                                ARIADNE1
Type options, press ENTER.
  2=Edit DBF  4=Delete    6=Link DBF  8=Elements  9=Attributes
/CUSTOMER_IMPORT/
Opt Elements                 File       Library    Field      Link Dbf Elm Atr
 _ CUSTOMER                  CUS_IMPORT *LIBL      *NONE                R   Y
```

Specify the database field/s

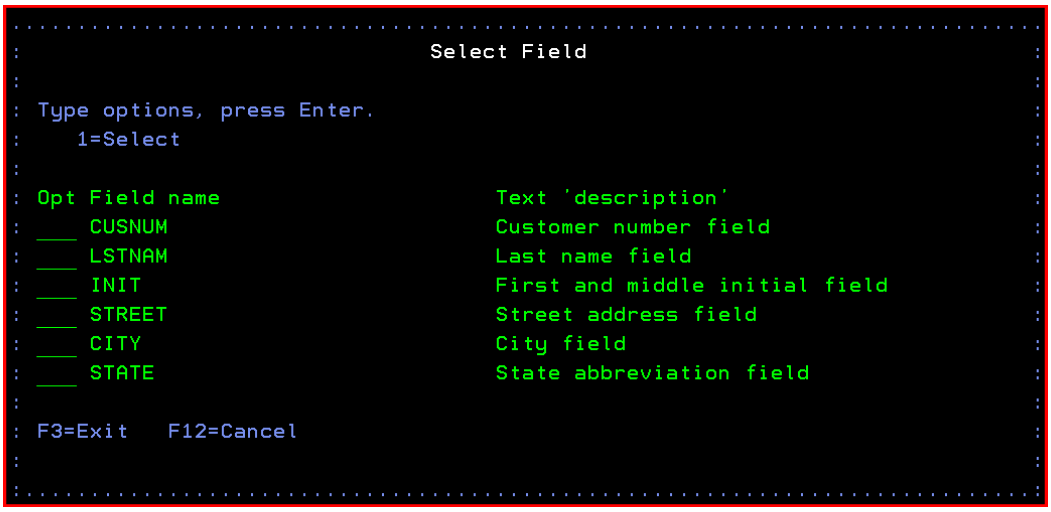
Now to map the data to the fields. Select the CUSTOMER element with 8=Elements. The field level elements will be displayed. Select CUSTOMER_NUMBER with 2=Edit DBF. You will notice that the file and Library are populated.



```
                   Work with XML-to-DBF map element - CHANGE

Map: CUSTOMER_IMPORT
/CUSTOMER_IMPORT/CUSTOMER/

Element Name               CUSTOMER_NUMBER

Action                     _             N/blank=No action, D=Data, R=Record
File Name                  CUS_IMPORT    Name, *NONE
Library                      *LIBL       Name
Field                      *SELECT       Name, *NONE, *SELECT
Data Type                                *ALPHA, *DATE, *NUMERIC, *TIME

Display Sequence                   15
```

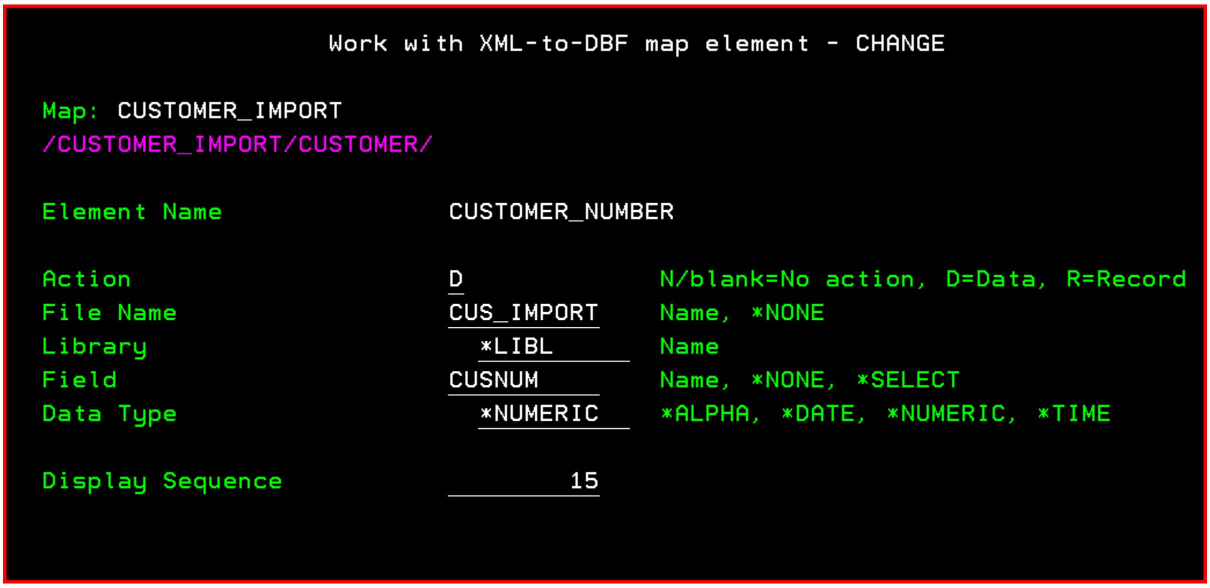
Type ‘D’ against Action to indicate the data from the element or attribute is to be mapped to the data base field specified.

Because Field as a value of *SELECT if you press ENTER, a list of the CUS_IMPORT file fields is displayed.



```
                              Select Field

Type options, press Enter.
   1=Select

Opt Field name                         Text 'description'
____ CUSNUM                            Customer number field
____ LSTNAM                            Last name field
____ INIT                              First and middle initial field
____ STREET                            Street address field
____ CITY                              City field
____ STATE                             State abbreviation field

F3=Exit   F12=Cancel
```

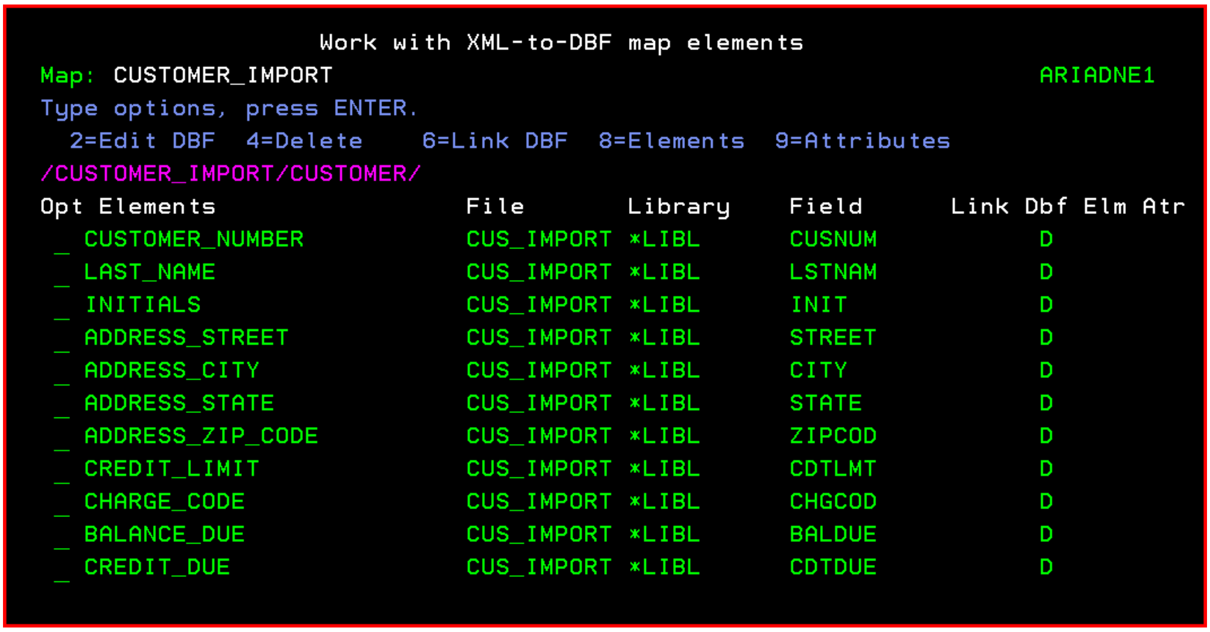
Select the field to be associated with the element, in this case select CUSNUM.



```
                       Work with XML-to-DBF map element - CHANGE

 Map: CUSTOMER_IMPORT
 /CUSTOMER_IMPORT/CUSTOMER/

 Element Name                  CUSTOMER_NUMBER

 Action                        D             N/blank=No action, D=Data, R=Record
 File Name                     CUS_IMPORT    Name, *NONE
 Library                         *LIBL       Name
 Field                         CUSNUM        Name, *NONE, *SELECT
 Data Type                       *NUMERIC    *ALPHA, *DATE, *NUMERIC, *TIME

 Display Sequence                        15
```

You will notice that the Data Type field is populated with the relevant value. Press ENTER to update.

Repeat this for each field on the database file. The result should be:



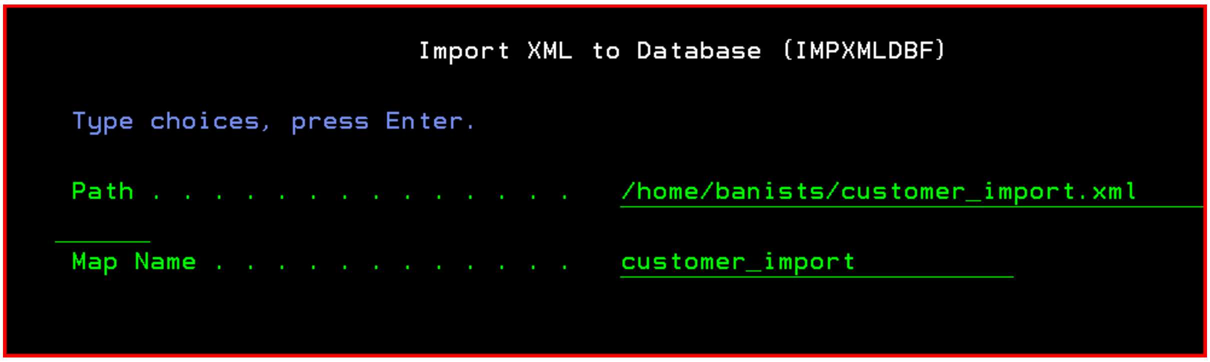
```
                       Work with XML-to-DBF map elements
 Map: CUSTOMER_IMPORT                                                    ARIADNE1
 Type options, press ENTER.
   2=Edit DBF   4=Delete    6=Link DBF   8=Elements   9=Attributes
 /CUSTOMER_IMPORT/CUSTOMER/
 Opt Elements                     File        Library     Field      Link Dbf Elm Atr
  _  CUSTOMER_NUMBER              CUS_IMPORT  *LIBL       CUSNUM          D
  _  LAST_NAME                    CUS_IMPORT  *LIBL       LSTNAM          D
  _  INITIALS                     CUS_IMPORT  *LIBL       INIT            D
  _  ADDRESS_STREET               CUS_IMPORT  *LIBL       STREET          D
  _  ADDRESS_CITY                 CUS_IMPORT  *LIBL       CITY            D
  _  ADDRESS_STATE                CUS_IMPORT  *LIBL       STATE           D
  _  ADDRESS_ZIP_CODE             CUS_IMPORT  *LIBL       ZIPCOD          D
  _  CREDIT_LIMIT                 CUS_IMPORT  *LIBL       CDTLMT          D
  _  CHARGE_CODE                  CUS_IMPORT  *LIBL       CHGCOD          D
  _  BALANCE_DUE                  CUS_IMPORT  *LIBL       BALDUE          D
  _  CREDIT_DUE                   CUS_IMPORT  *LIBL       CDTDUE          D
```

The map is now complete.

Step 3 - Import XML into the IBM i database

Using menu option 3 Extract data from XML to database file, the prompt for the command IMPXMLDBF (Import XML to Database) is displayed.

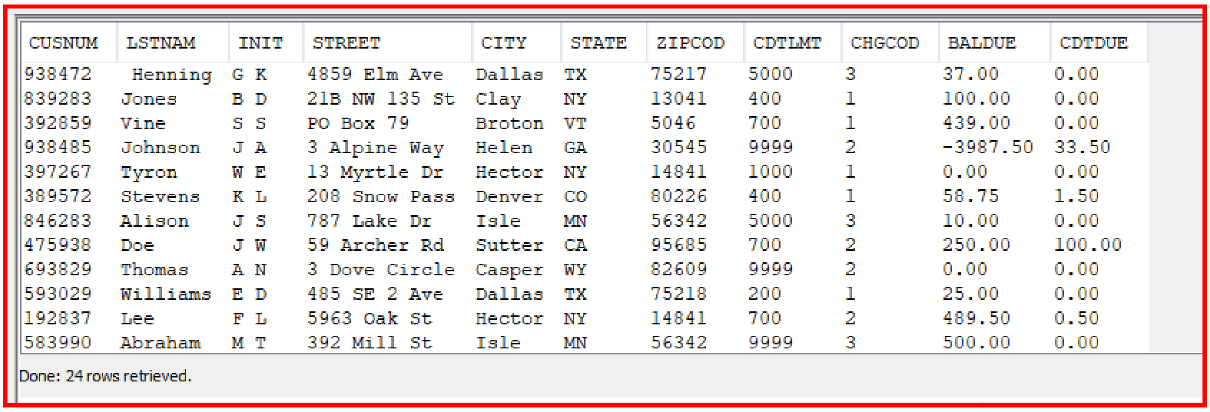
Enter the full path to the XML document to be imported and Map Name to be used.
Note: in this example, I am using the same XML file used to initially generate the map. A different XML document can be specified but the structure of the XML must be identical to XML document used to generate the map.



```
                      Import XML to Database (IMPXMLDBF)

 Type choices, press Enter.

 Path . . . . . . . . . . . . . .   /home/banists/customer_import.xml

 Map Name . . . . . . . . . . . .   customer_import
```

On completion, **XML document /home/banists/customer_import.xml imported successfully**

The database file:



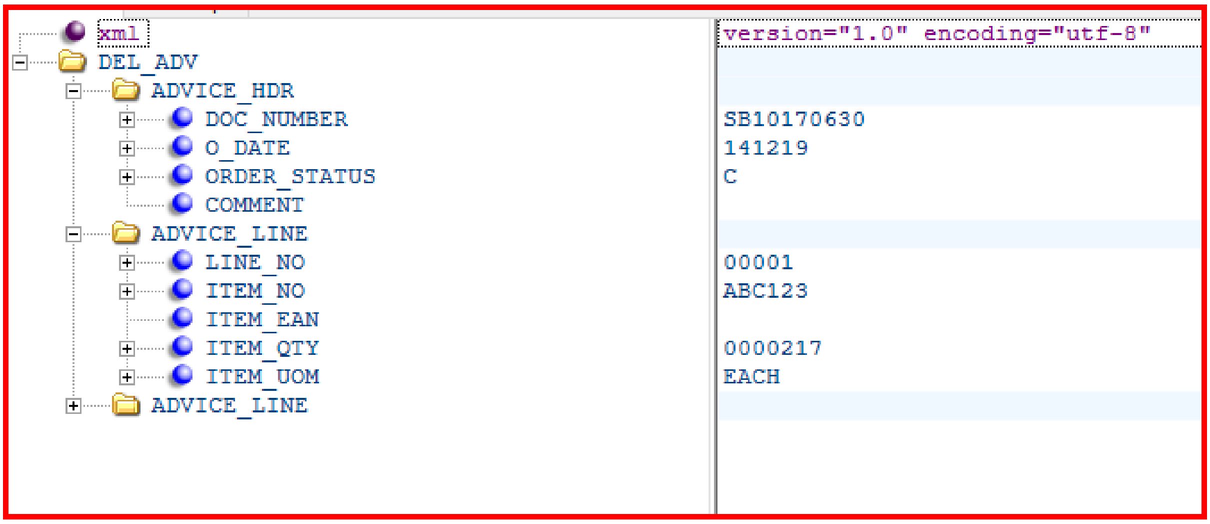
CUSNUM	LSTNAM	INIT	STREET	CITY	STATE	ZIPCOD	CDTLMT	CHGCOD	BALDUE	CDTDUE
938472	Henning	G K	4859 Elm Ave	Dallas	TX	75217	5000	3	37.00	0.00
839283	Jones	B D	21B NW 135 St	Clay	NY	13041	400	1	100.00	0.00
392859	Vine	S S	PO Box 79	Broton	VT	5046	700	1	439.00	0.00
938485	Johnson	J A	3 Alpine Way	Helen	GA	30545	9999	2	-3987.50	33.50
397267	Tyron	W E	13 Myrtle Dr	Hector	NY	14841	1000	1	0.00	0.00
389572	Stevens	K L	208 Snow Pass	Denver	CO	80226	400	1	58.75	1.50
846283	Alison	J S	787 Lake Dr	Isle	MN	56342	5000	3	10.00	0.00
475938	Doe	J W	59 Archer Rd	Sutter	CA	95685	700	2	250.00	100.00
693829	Thomas	A N	3 Dove Circle	Casper	WY	82609	9999	2	0.00	0.00
593029	Williams	E D	485 SE 2 Ave	Dallas	TX	75218	200	1	25.00	0.00
192837	Lee	F L	5963 Oak St	Hector	NY	14841	700	2	489.50	0.50
583990	Abraham	M T	392 Mill St	Isle	MN	56342	9999	3	500.00	0.00

Done: 24 rows retrieved.

More complex XML structures

Some XML structures may include elements for Header data and repeated Detail data. For example, Order header and Order Lines.

Here is a basic Delivery advice structure.

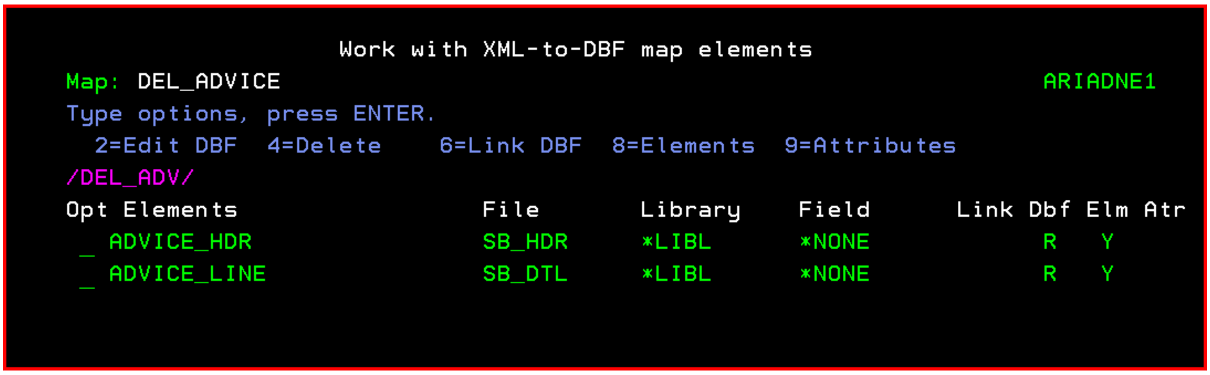


```
xml                     version="1.0" encoding="utf-8"
DEL_ADV
    ADVICE_HDR
        DOC_NUMBER      SB10170630
        O_DATE          141219
        ORDER_STATUS    C
        COMMENT
    ADVICE_LINE
        LINE_NO         00001
        ITEM_NO         ABC123
        ITEM_EAN
        ITEM_QTY        0000217
        ITEM_UOM        EACH
    ADVICE_LINE
```

Link DBF

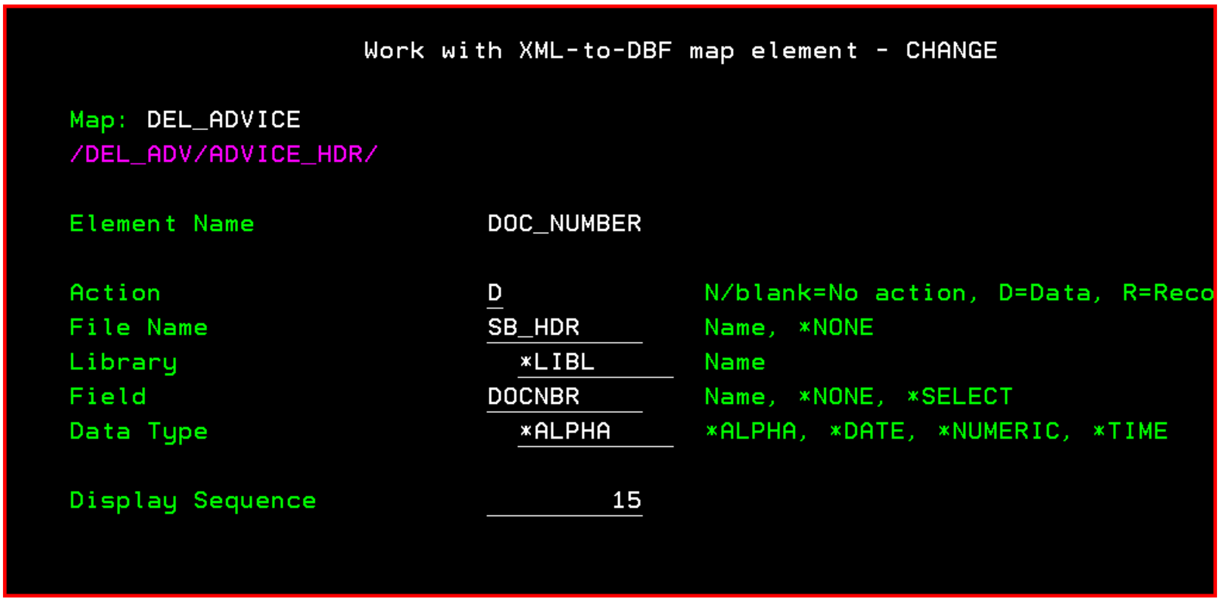
You can see the Advice Header element includes DOC_NUMBER which identifies the transaction. But the Advice Line element does not include the DOC_NUMBER.

After specifying which database files will be updated during import, the map would appear as:



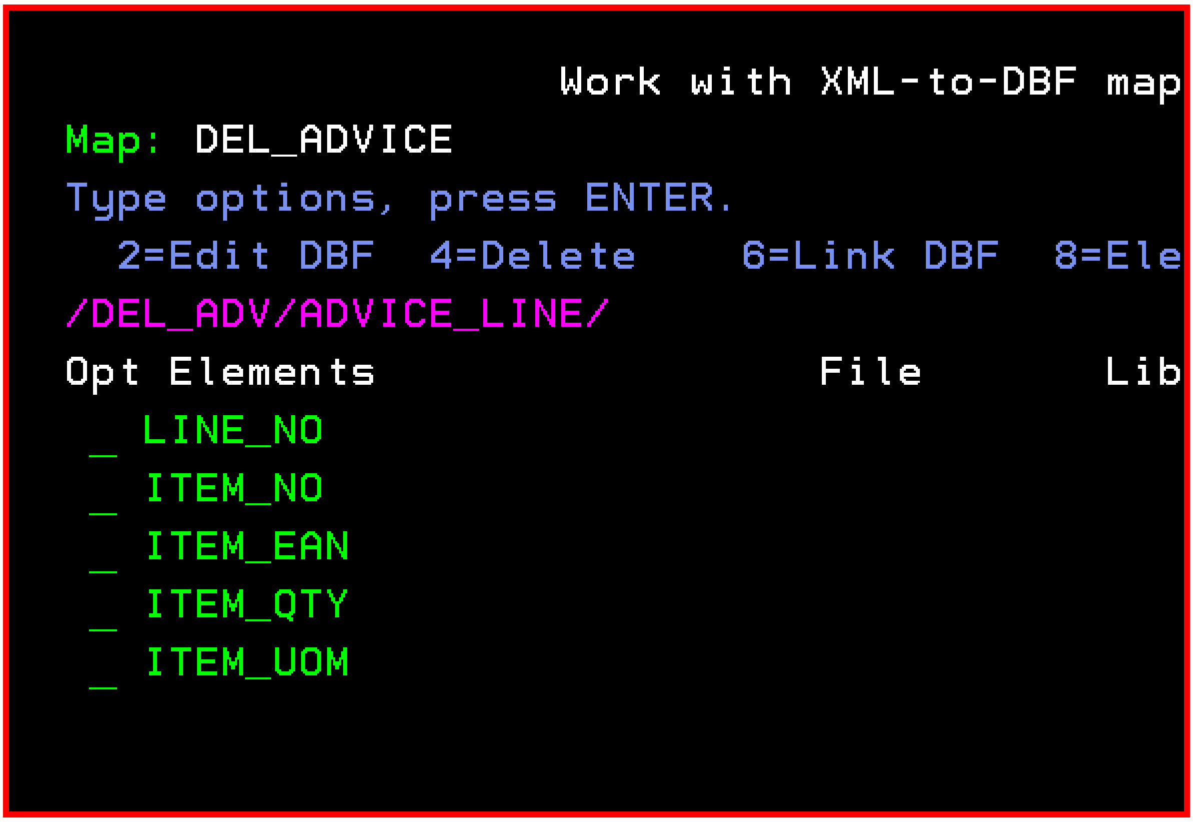
```
                       Work with XML-to-DBF map elements
 Map:  DEL_ADVICE                                                         ARIADNE1
 Type options, press ENTER.
   2=Edit DBF   4=Delete    6=Link DBF   8=Elements   9=Attributes
 /DEL_ADV/
 Opt Elements                    File        Library     Field       Link Dbf Elm Atr
  _  ADVICE_HDR                  SB_HDR      *LIBL       *NONE            R    Y
  _  ADVICE_LINE                 SB_DTL      *LIBL       *NONE            R    Y
```

DOC_NUMBER should be associated to the Header as previously explained, for example:



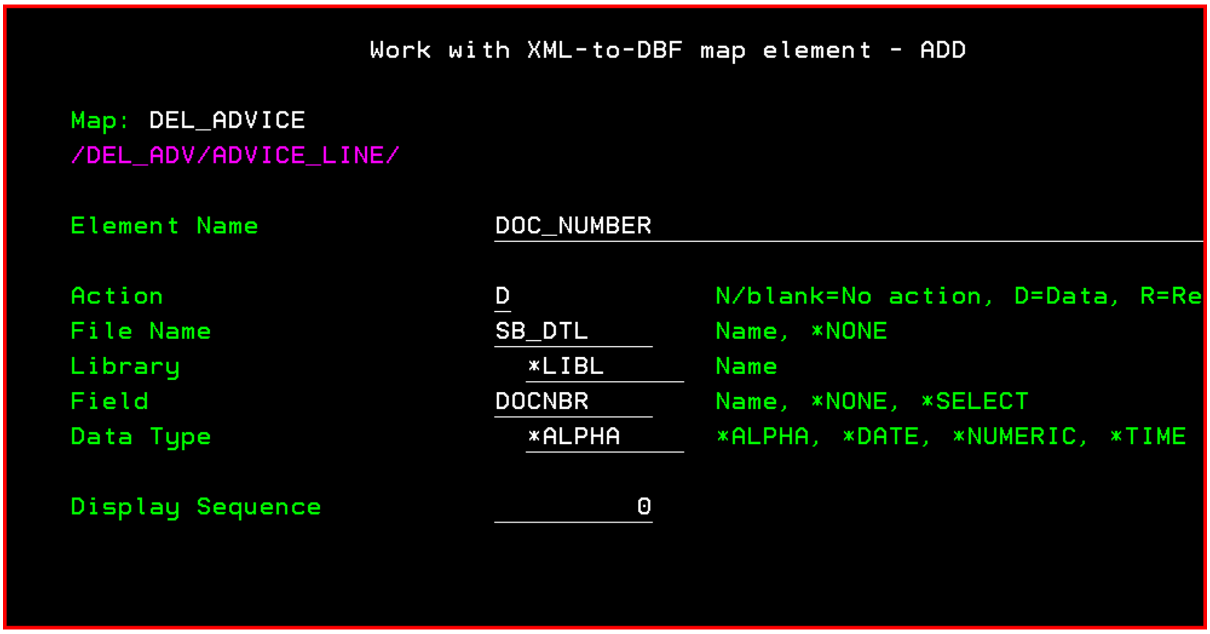
```
                    Work with XML-to-DBF map element - CHANGE

 Map: DEL_ADVICE
 /DEL_ADV/ADVICE_HDR/

 Element Name                DOC_NUMBER

 Action                      D             N/blank=No action, D=Data, R=Reco
 File Name                   SB_HDR        Name, *NONE
 Library                       *LIBL       Name
 Field                       DOCNBR        Name, *NONE, *SELECT
 Data Type                     *ALPHA      *ALPHA, *DATE, *NUMERIC, *TIME

 Display Sequence                     15
```

To populate DOCNBR field when populating the Advice Detail file:



```
                    Work with XML-to-DBF map
Map: DEL_ADVICE
Type options, press ENTER.
  2=Edit DBF   4=Delete    6=Link DBF   8=Ele
/DEL_ADV/ADVICE_LINE/
Opt Elements                     File        Lib
 _   LINE_NO
 _   ITEM_NO
 _   ITEM_EAN
 _   ITEM_QTY
 _   ITEM_UOM
```

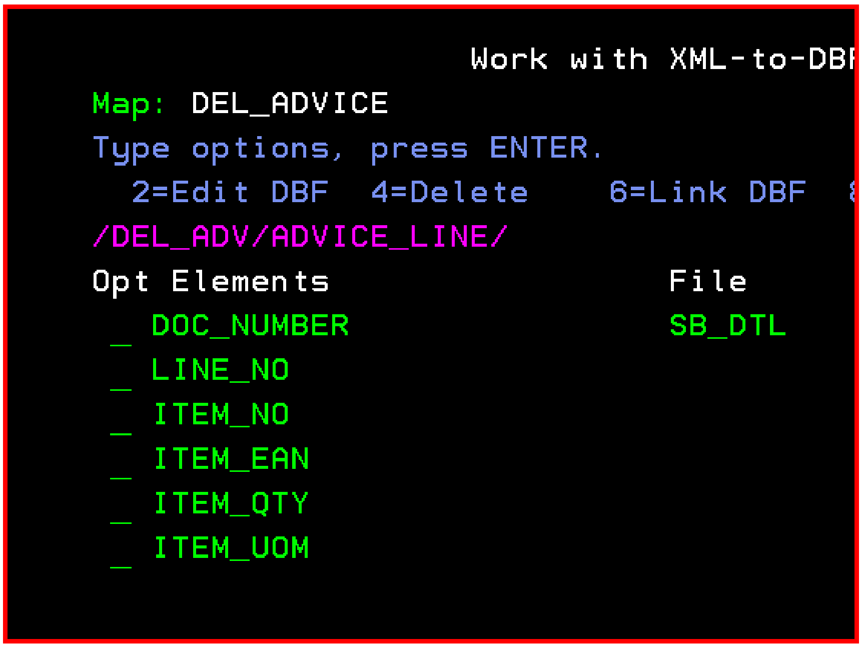
1. Press F6=Add to create a new element.
   Enter a name for the Element and the relevant database field information



```
                    Work with XML-to-DBF map element - ADD

  Map: DEL_ADVICE
  /DEL_ADV/ADVICE_LINE/

  Element Name                DOC_NUMBER

  Action                      D             N/blank=No action, D=Data, R=Re
  File Name                   SB_DTL        Name, *NONE
  Library                       *LIBL       Name
  Field                       DOCNBR        Name, *NONE, *SELECT
  Data Type                     *ALPHA      *ALPHA, *DATE, *NUMERIC, *TIME

  Display Sequence                     0
```

and press ENTER.

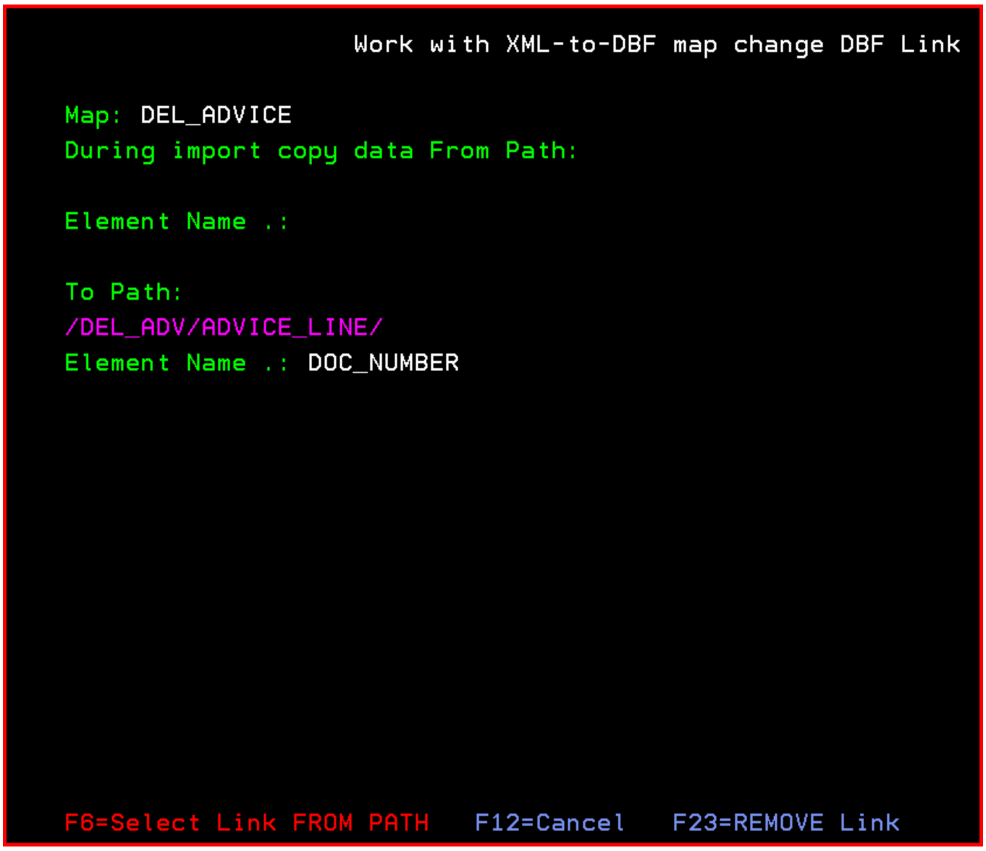
The ADVICE_LINE element structure will now include the DOC_NUMBER sub-element.



```
                    Work with XML-to-DBF
  Map: DEL_ADVICE
  Type options, press ENTER.
    2=Edit DBF  4=Delete   6=Link DBF  8
  /DEL_ADV/ADVICE_LINE/
  Opt Elements                File
   _  DOC_NUMBER              SB_DTL
   _  LINE_NO
   _  ITEM_NO
   _  ITEM_EAN
   _  ITEM_QTY
   _  ITEM_UOM
```

2. To populate the new DOC_NUMBER element during import, select the element with 6=Link DBF

The Change DBF Link screen will be displayed.

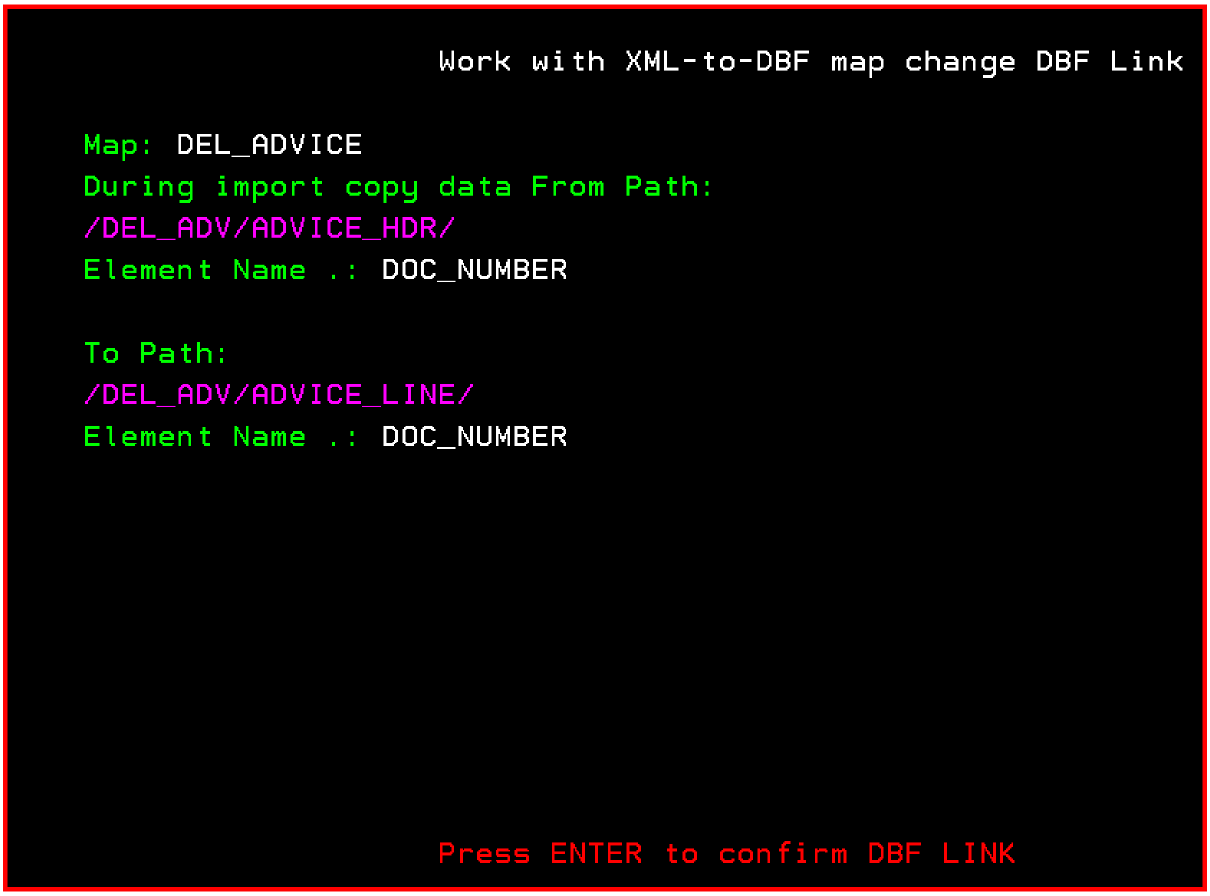


```
                           Work with XML-to-DBF map change DBF Link

   Map: DEL_ADVICE
   During import copy data From Path:

   Element Name . :

   To Path:
   /DEL_ADV/ADVICE_LINE/
   Element Name . : DOC_NUMBER









   F6=Select Link FROM PATH   F12=Cancel   F23=REMOVE Link
```

3. Press F6=Select Link FROM PATH.
   The top level of the XML document will be displayed. In this example DEL_ADV.

4. Use subfile option 8=Elements to locate the ADVICE_HDR element DOC_NUMBER.
5. Select ADVICE_HDR DOC_NUMBER with option 1=Select

The change DBF Link screen will be re-displayed showing the FROM and TO paths to be linked.



```
                       Work with XML-to-DBF map change DBF Link

    Map: DEL_ADVICE
    During import copy data From Path:
    /DEL_ADV/ADVICE_HDR/
    Element Name .: DOC_NUMBER

    To Path:
    /DEL_ADV/ADVICE_LINE/
    Element Name .: DOC_NUMBER










                       Press ENTER to confirm DBF LINK
```

6. Press ENTER to confirm the creation of the DBF link.

   The ADVICE_LINE element structure will be re-displayed.

Notes:

Against the new DOC_NUMBER element, in the Link column ‘TO’ is displayed to indicate that during the Import process data will be copied to this element from the associated Linked element.

If you review the ADVICE_HDR element structure against the DOC_NUMBER element, you will see ‘FROM’ under Link. This indicates it is the source of a DBF LINK and that data will be copied from this element to the associated Linked element/s during import.
When 6=Link DBF is entered against ADVICE_HDR/DOC_NUMBER element, a list of Linked To Path/Elements is displayed.

DBF Links can only maintained via the ‘TO’ element