BPA
Platform

Technical Overview

Import Flat File Tool

codeless
PLATFORMS

Copyright

Head Office:

Codeless Platforms

Suite 1 & 2 Bourne Gate

25 Bourne Valley Road

Poole

BH12 1DY

United Kingdom

Tel: +44 (0) 330 99 88 700

Email: enquiries@codelessplatforms.com

Trademarks

The information provided in this publication may contain errors, omissions, or typographical errors or may be out of date. Orbis Software Ltd T/A Codeless Platforms may change, delete, or update any published information at any time and without prior notice. The information published in this document is provided for informational purposes only and is not binding on Orbis Software Ltd T/A Codeless Platforms in any way except to the extent that it is specifically indicated to be so.

Contents

Introduction

The **Import Flat File** tool imports a delimited or fixed-width file from disk into BPA Platform, extracts the file content including any file properties, then outputs it as either a recordset or as XML.

The columns that make up the recordset, and the nodes and nested elements for the XML file, can either be inferred from an imported example or created manually.

Data validation can also be applied to only import required data.

What is a Flat File?

A flat file (also referred to as a "flat file database") stores unstructured data in plain text format. It does not contain:

- Any internal hierarchy
- Any links to other files
- Any word processing information or formatting

For BPA Platform, a flat file contains a single "table" of data, with one record per line. You specify the data type of each column and use delimiters to separate each column in a record.

Features

The **Import Flat File** tool provides the following features

- Manually create a schema for the XML output consisting of nodes and nested elements
- Configure properties of each node and element to select the data to be extracted for each
- Preview the XML schema structure before saving and running the task
- Specify the source file properties to be extracted at task run-time and exposed to other steps
- Browse and select a file containing column headings in the first row to structure the output data
- Output XML for use by other task steps
- Output XML as a string for use by other task steps
- Output a recordset for use by other task steps

Working with other Tools

The **Import Flat File** tool can directly interact with the following tools:

Consuming from Other Tools

The **Import Flat File** tool can directly consume objects outputted by the following tools:

Icon	Tool Name	Tool Category
	Import Flat File	Input
xml	**Import XML Document**	Input

Objects Consumed

The following objects, outputted by the above tools, can be directly consumed by the **Import Flat File** tool:

- **Recordset** — Tabular data from any BPA Platform tool capable of exposing such data (see above)

Exposing to Other Tools

The **Import Flat File** tool exposes objects which can be directly consumed by the following tools:

Icon	Tool Name	Tool Category
xml	**Import XML Document**	Input
</>	**Convert Recordset to XML**	Format
✗✓	**Create Workflow Job**	Format
	Format as Flat File	Format

Icon	Tool Name	Tool Category
	Format as HTML	Format
	Format as HTML Pro	Format
	Format as Text	Format
	Run Crystal Report	Format
	Run Microsoft Reporting Services	Format
	Run Microsoft Word (Merge)	Format
	Save File	Output
	Transfer File (FTP)	Output
	Call COM Object	Execute
	Run External Program	Execute
	File Management	General
	Filter Data	General

Objects Exposed

The **Import Flat File** exposes the following objects which can be directly consumed by the above tools:

- **Step Properties** — Standard step properties are available allowing you to use statistical data of the tool

- **InputData** — This object is only available if a file is consumed from a previous task step. It contains the imported file contents.
- **FileProperties** — This object exposes the source file's `Name`, `Size`, `Type`, `DateModified`, `DateCreated`, `DateAccessed`, and `Attributes` properties collected at task run-time. You can set which properties are exposed in the File Properties dialog box.
- **ImportedFileDetails** — This object exposes the imported file's **Path** and **Filename**.

NOTE: FileProperties and **ImportedFileDetails** are updated each time the task step processes a file. If a wildcard (`*`) is used to select multiple files in the General tab, the objects only contain values for the last file processed; the previous file's information being overwritten.

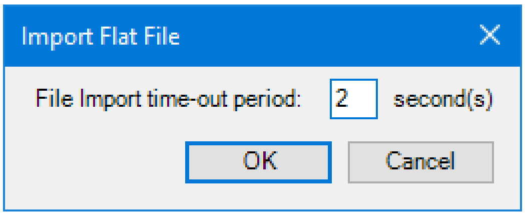
- **Recordset** — Tabular data from any BPA Platform tool capable of exposing such data (see above) (This property is only available if a **Recordset** output is selected (see Specifying the Output Type).
- **OutputXML** — This object exposes two sub-objects (this property is only available if **XML** output is selected (see Specifying the Output Type)):
  - **Output Schema** — This contains the XSD file for the output
  - **Output XML** — This contains the XML produced by the **Import XML Document** tool
- **RowCount** — Returns the number of rows processed for the output. This applies to both recordset and XML outputs, and considers whether a header row was included in the consumed file (input) or whether empty rows are ignored — see Specifying How to Read the Imported File.
- **SuccessRowCount** — Returns the number of rows successfully processed for the output. This applies to both recordset and XML outputs. **SuccessRowCount** + **FailRowCount** should always equal **RowCount**.
- **FailRowCount** — Returns the number of rows that the **Import Flat File** tool was unable to process. This applies to both recordset and XML outputs.

Setting the Import Time Out

You can set the time out period for how long the **Import Flat File** tool attempts to import a file. If the file cannot be imported, after the expiry time, an error is written to the Event Log.

You open the **Import Flat File** time out dialog box by either:

You open this window from the resources tree — expand ***System* > *Tools* > *Input*** and double-click ***Import Flat File*** in the items list.



File Import time-out period: 2 second(s)

Set the import time-out period; the default is 2 seconds.

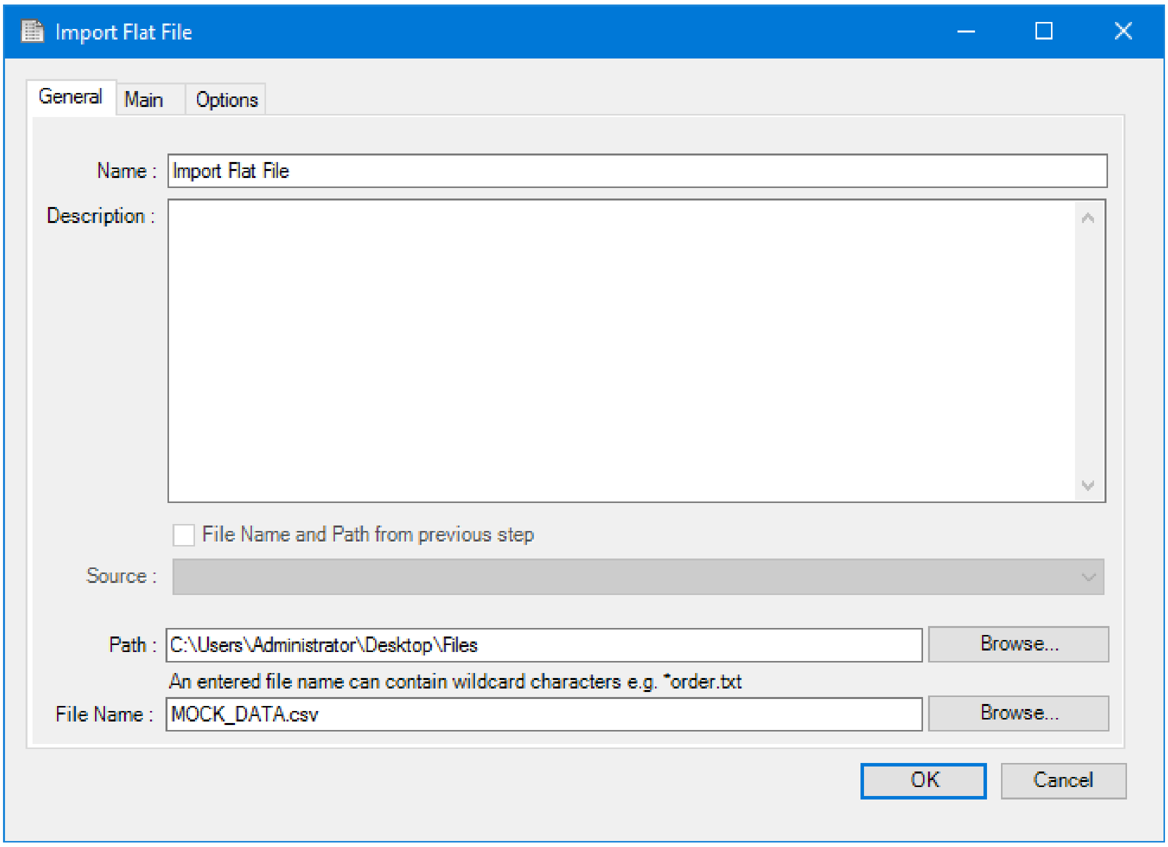
Step Configuration

To add a new **Import Flat File** step to an existing task, you either:

- Click and drag the **Import Flat File** icon from the **Task Browser** to the task **Design** area.
- From the task's **Design** tab, right-click on empty space and select ***New* > *Input* > *Import Flat File.***

For a detailed description of how to create new tasks, refer to the product help.

About the General Tab



Name : Import Flat File

Description :

□ File Name and Path from previous step

Source :

Path : C:\Users\Administrator\Desktop\Files

An entered file name can contain wildcard characters e.g. *order.txt

File Name : MOCK_DATA.csv

The **General** tab is used to enter the following details for the step:

- **Name** — Enter a meaningful name for the step

TIP: If this task instance makes use of two or more **Import Flat File** steps, ensure the **Name** used is unique for each individual step.

- **Description** — If required, enter a description of this step
- **File name and path from a previous step** — Enable this option to import a document previously created or imported by another task step
  - **Source** — Contains all available task steps that have available document outputs
- **Path** — The full folder path to the location of the document

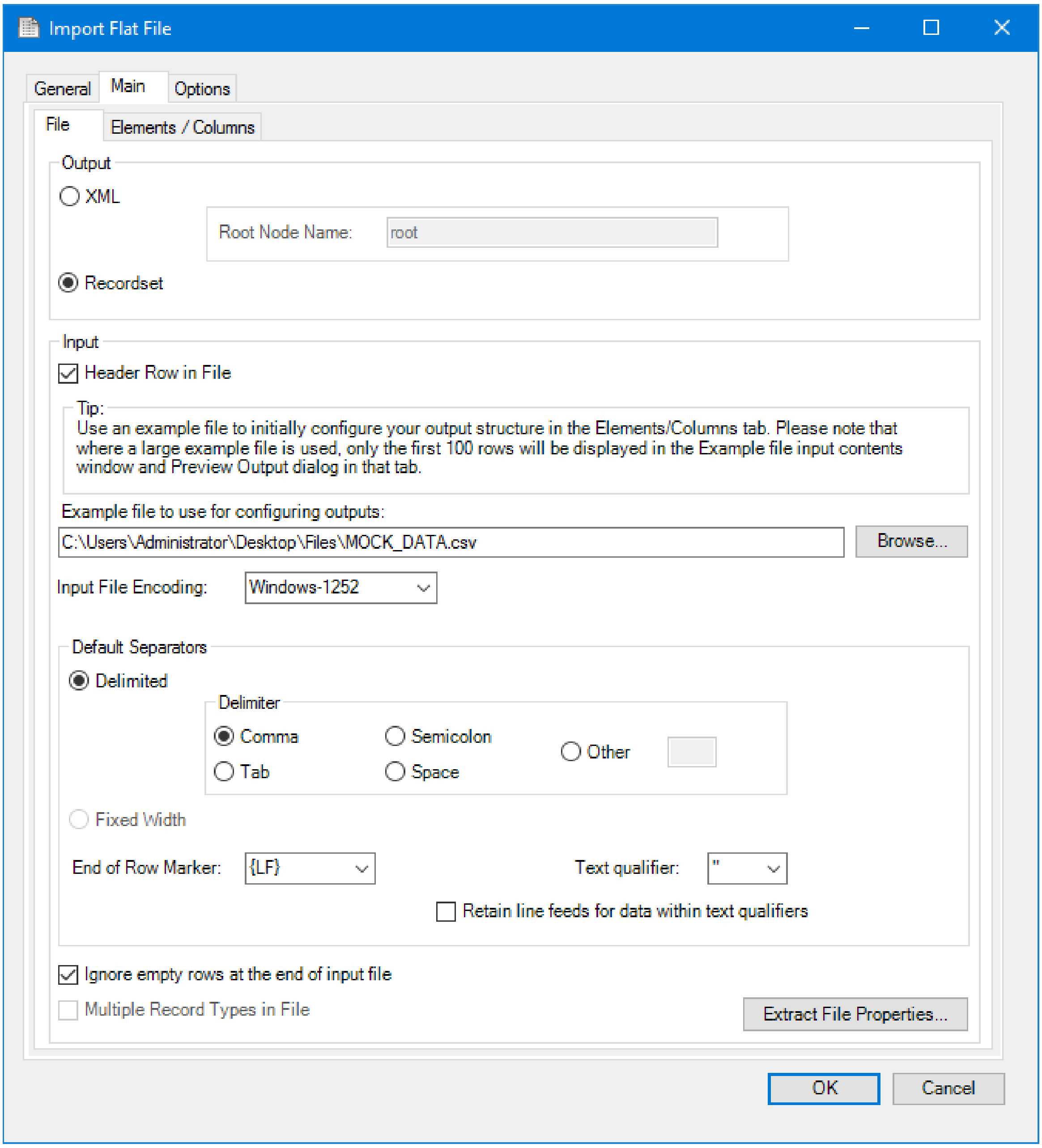
NOTE: Ensure the logged in Windows user has read access to the folder specified here.

- **File Name** — The document to be imported

> **TIP:** Use of wildcards are supported here so that multiple documents can be imported by the same step at run-time.

About the Main Tab

The **Main** tab is used to select the output type required and also how data is to be extracted from the source file.



Output

○ XML

Root Node Name: root

● Recordset

Input

☑ Header Row in File

Tip:
Use an example file to initially configure your output structure in the Elements/Columns tab. Please note that where a large example file is used, only the first 100 rows will be displayed in the Example file input contents window and Preview Output dialog in that tab.

Example file to use for configuring outputs:
C:\Users\Administrator\Desktop\Files\MOCK_DATA.csv

Input File Encoding: Windows-1252

Default Separators

● Delimited

Delimiter

● Comma ○ Semicolon ○ Other
○ Tab ○ Space

○ Fixed Width

End of Row Marker: {LF}

Text qualifier: "

□ Retain line feeds for data within text qualifiers

☑ Ignore empty rows at the end of input file

□ Multiple Record Types in File

Specifying the Output Type

You can choose between:

- **XML**
    - **Root Node Name** — You must provide the name of the root (top) node in your XML.
- **Recordset**

Specifying How to Read the Imported File

In the **Input** pane, specify how the imported file is read and the data extracted:

- **Header Row in File** — If the imported file contains a header row, enable this parameter. Without this, the first row in your file is treated as contents.
- **Example file to use for configuring outputs** — If you have an example file of the expected output from this task step, enter the file path and name here. The **Import Flat File** tool uses certain features of this file to auto-populate parameters and also to provide a preview of the expected output when building the structure.
- **Input File Encoding** — If known, select the encoding. You can also manually enter the encoding as an ANSI code page identifier. If an **Example file to use for configuring outputs** has been selected, the encoding is detected and displayed here.
- You must specify the **Default Separators** for the imported file:
    - **Delimited** — Use this option if the fields in the imported file are separated by a character. Select the relevant **Delimiter**, or if **Other** specify the delimiter in the box provided.
    - **Fixed Width** — Use this option if the fields in the imported file have a set width. Note that if **Header Row in File** is enabled then **Fixed Width** is not available as header rows do not tend to appear in fixed width files.
- **End of Row Marker** — If known, select the **End of Row Marker**. If an **Example file to use for configuring outputs** has been selected, the end of marker is automatically selected.
- **Text qualifier** — This is a symbol denoting the beginning and end of a text string. These are particularly important if a field in the imported file contains the delimiter symbol as a standard character, for . If the imported file contains text qualifiers, choose the relevant one from the drop-down.
- **Retain line feeds for data within text qualifiers** — In a similar vein, if fields in the imported file contain line feeds in the text string, such as addresses where each address line is a new line, enable **Retain line feeds for data within text qualifiers**. This keeps the line markers in place when processing the file.

- **Ignore empty rows at the end of the input file** — Some input files may contain a blank row as the last line. If required, select this option to *not* add blank rows to the XML or recordset output should any be found.
- **Multiple Record Types in File** — Only used if outputting an XML file. A record type is a tag at the beginning of the data row, used to indicate a new type of record in the same file.

  The record types can have a parent-child relationship or can be at the same level.

  At the time of writing, recordsets do not support parent-child relationships, however, such multi-type files can be outputted as XML. Enable this parameter if multiple record type support is required in the XML output.

Extracting File Properties

If required, you can specify which of the imported document's file properties are made available in the Task Browser at run-time:

- Name
- Size
- Item type
- Date modified
- Date created
- Date accessed
- Attributes

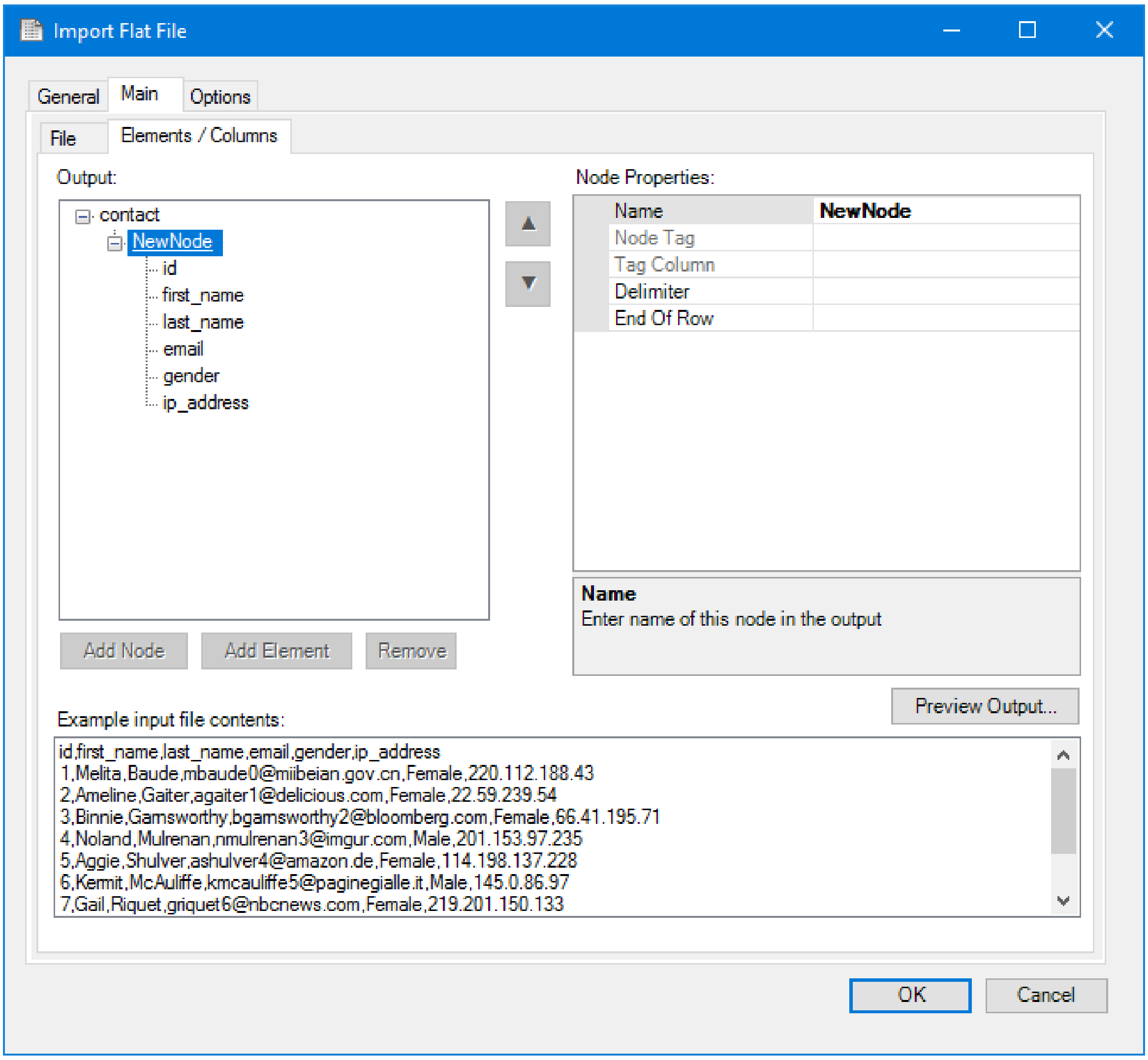
Click **Extract File Properties**.

Defining the Output Structure

You define the output structure in the **Elements / Columns** tab.

Creating XML Output

When you create output and have indicated that your input file contains a header row, the **Import Flat File** tool creates a basic structure based on the header row:



Output:

```
⊟ contact
  ⊟ NewNode
      id
      first_name
      last_name
      email
      gender
      ip_address
```

Node Properties:

Name	NewNode
Node Tag	
Tag Column	
Delimiter	
End Of Row	

Name
Enter name of this node in the output

Example input file contents:

```
id,first_name,last_name,email,gender,ip_address
1,Melita,Baude,mbaude0@miibeian.gov.cn,Female,220.112.188.43
2,Ameline,Gaiter,agaiter1@delicious.com,Female,22.59.239.54
3,Binnie,Gamsworthy,bgamsworthy2@bloomberg.com,Female,66.41.195.71
4,Noland,Mulrenan,nmulrenan3@imgur.com,Male,201.153.97.235
5,Aggie,Shulver,ashulver4@amazon.de,Female,114.198.137.228
6,Kemit,McAuliffe,kmcauliffe5@paginegialle.it,Male,145.0.86.97
7,Gail,Riquet,griquet6@nbcnews.com,Female,219.201.150.133
```

Without a header row in the input file, only the root node and a single child node are created. Use **Add Node** and **Add Element** to create the required XML structure.

For both automatically and manually defined structures, you must also define the properties:

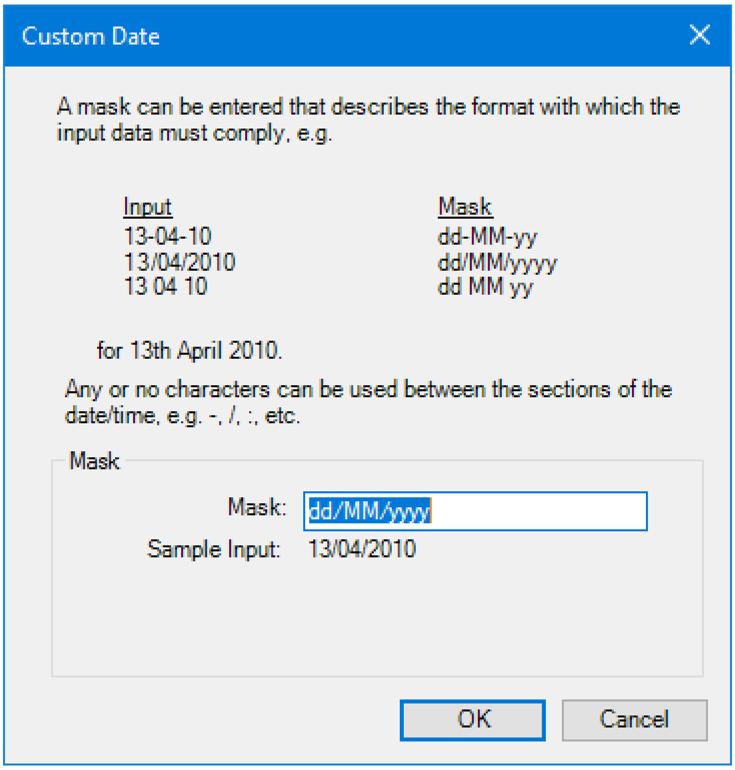
- **Node Properties**
  - □ **Name** — The name of the XML node.
  - □ **Node Tag** — If creating a multiple record type file, enter the tag string of the record types for this node.

- **Tag Column / Position** — If the input file is delimited, this displays as **Tag Column** and is the column name where the tag occurs. If it is a fixed width file, this displays as **Tag Position** and is the character position where the tag starts.
- **Delimiter** — If inputting a delimited file and this particular node has a different delimiter, choose the correct one here.
- **End of Row Character** — If this particular node has a different end of row character, choose the correct one here.

Element Properties

- **Name** — The name of the XML element
- **Type** — The data type of the XML element. The default value is `string`.
- **Max Characters** — By default, the maximum number of characters for the element is `100`. Adjust this as required. Note that if a **Date**, **DateTime**, or **Time** data type has been selected then this property is not editable as the length is set automatically to match the selected format.
- **Format** — The data format of the XML element. The default value is `Unformatted` in which case no validation takes place. If required, you can enter your own custom format.

  If **Type** is either `Date`, `DateTime`, or `Time` and a `CUSTOM` format is selected, you are prompted to enter a **Mask** to describe the format, for example:



A mask can be entered that describes the format with which the input data must comply, e.g.

Input	Mask
13-04-10	dd-MM-yy
13/04/2010	dd/MM/yyyy
13 04 10	dd MM yy

for 13th April 2010.

Any or no characters can be used between the sections of the date/time, e.g. -, /, :, etc.

Mask: dd/MM/yyyy

Sample Input: 13/04/2010

- **Required** — This property indicates whether a value is always required for this element at task run-time. Selecting **Yes** may cause the step to fail if a value is not present; **No** (default) and the step continues even when the element is empty.

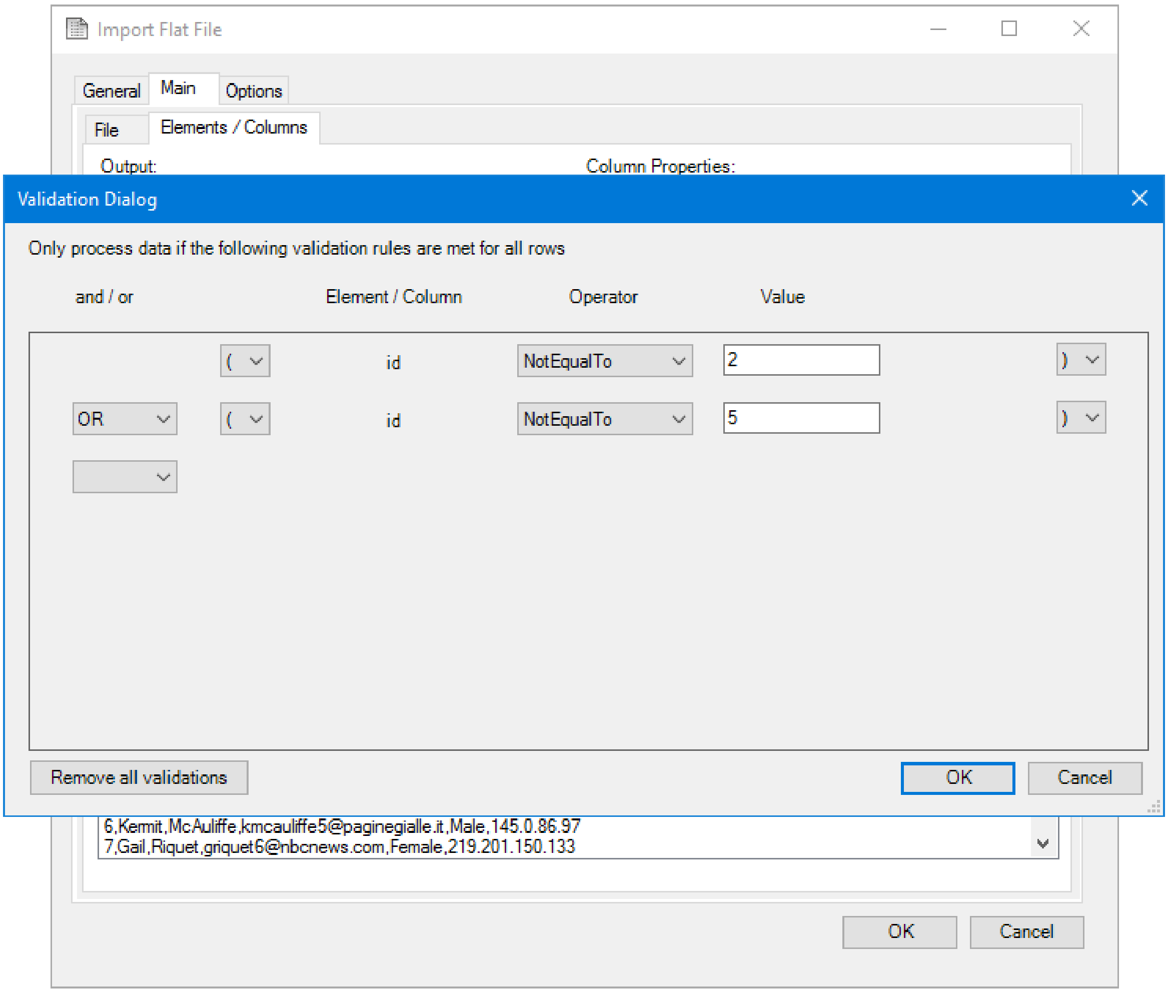
- □ **From** — If the input file is of fixed width format, this property states when the value for this element starts.
- □ **To** — If the input file is of fixed width format, this property states when the value for this element ends.
- □ **Validations** — Any data validation rules applied to this element — see below.

- **Preview Output** — Use this to check how the output is structured

- **Example input file contents** — If an example file has been defined in the File tab, this displays a small extract of that file as an aid to building your output

Validating Data

For the **Import Flat File's** output, you can create one or more logical expression rules to filter the data contained in the input file(s) at task run time. This allows you to capture and correct any errors at the earliest possible time in a task.



Only process data if the following validation rules are met for all rows

and / or		Element / Column	Operator	Value	
	(	id	NotEqualTo	2	)
OR	(	id	NotEqualTo	5	)

```
6,Kermit,McAuliffe,kmcauliffe5@paginegialle.it,Male,145.0.86.97
7,Gail,Riquet,griquet6@nbcnews.com,Female,219.201.150.133
```

In the example above, all records in the input file are processed apart from those with 2 or 5 as their ID. Another important example would be to ignore rows that have blank fields.

If implementing multiple rules, ensure you build your rules in a logical sequence.

NOTE: Nested rules are not supported. Nor can you create rules which involve multiple elements or columns.

Supported **Operators** include:

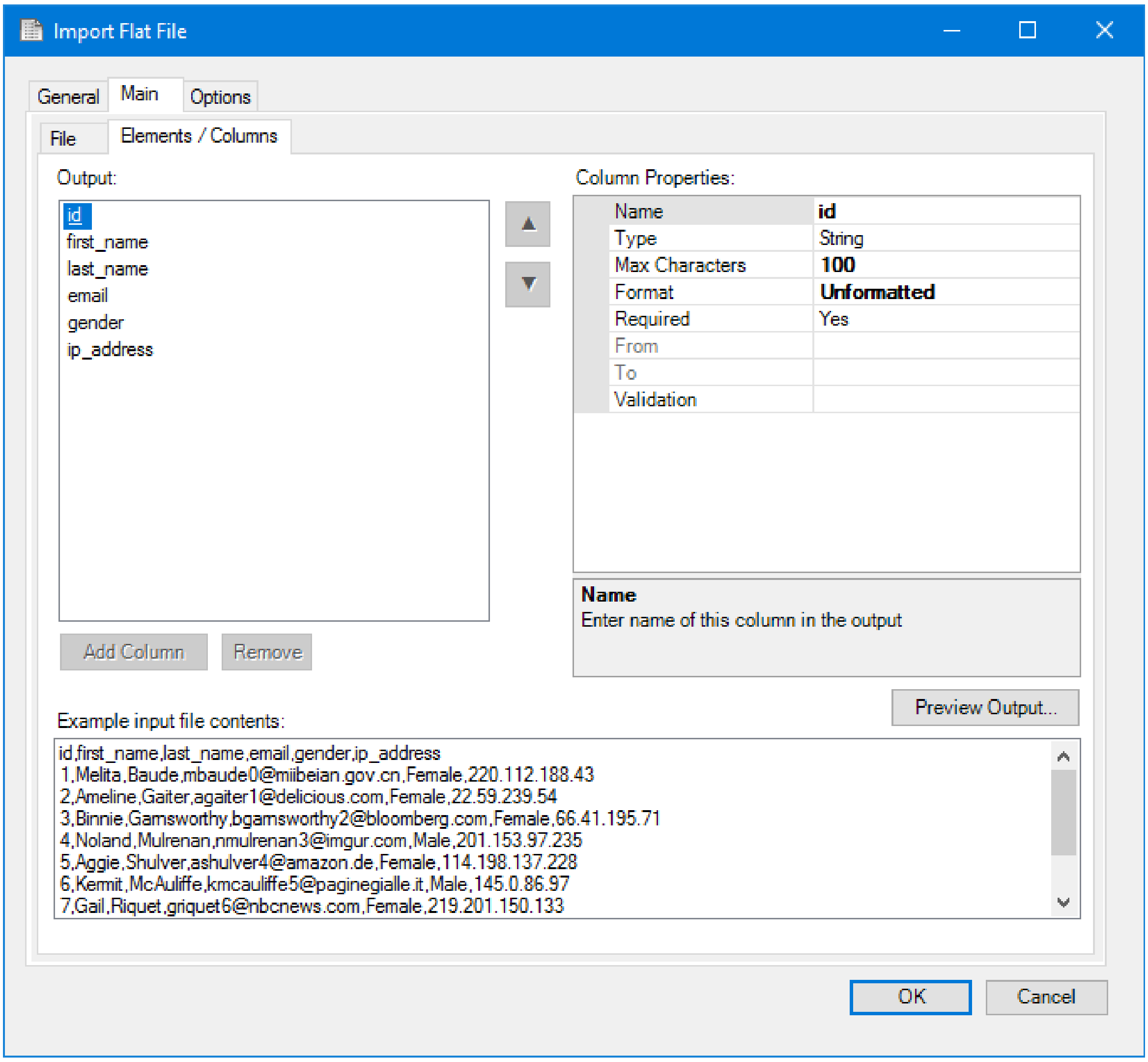
- Equals
- Starts With
- Contains
- Ends With
- Greater Than
- Less Than
- Between
- Not Equal To
- Not Starting With
- Not Containing
- Not Ending With
- Is Not Between
- Is One Of
- Is Not One Of
- Is Null
- Is Not Null

NOTE: If **Import Flat File** is outputting XML (see Creating XML Output), then any rules are applied against the complete dataset in one operation at task run-time.

If outputting recordsets (see Creating Recordset Output), then rules are applied against each record in the recordset as it is processed by the subsequent consuming step at task run-time.

Creating Recordset Output

When you create output and have indicated that your input file contains a header row, the **Import Flat File** tool creates a basic structure based on the header row:



Output:

id
first_name
last_name
email
gender
ip_address

Column Properties:

Name	**id**
Type	String
Max Characters	**100**
Format	**Unformatted**
Required	Yes
From	
To	
Validation	

Name
Enter name of this column in the output

Example input file contents:

```
id,first_name,last_name,email,gender,ip_address
1,Melita,Baude,mbaude0@miibeian.gov.cn,Female,220.112.188.43
2,Ameline,Gaiter,agaiter1@delicious.com,Female,22.59.239.54
3,Binnie,Gamsworthy,bgamsworthy2@bloomberg.com,Female,66.41.195.71
4,Noland,Mulrenan,nmulrenan3@imgur.com,Male,201.153.97.235
5,Aggie,Shulver,ashulver4@amazon.de,Female,114.198.137.228
6,Kermit,McAuliffe,kmcauliffe5@paginegialle.it,Male,145.0.86.97
7,Gail,Riquet,griquet6@nbcnews.com,Female,219.201.150.133
```

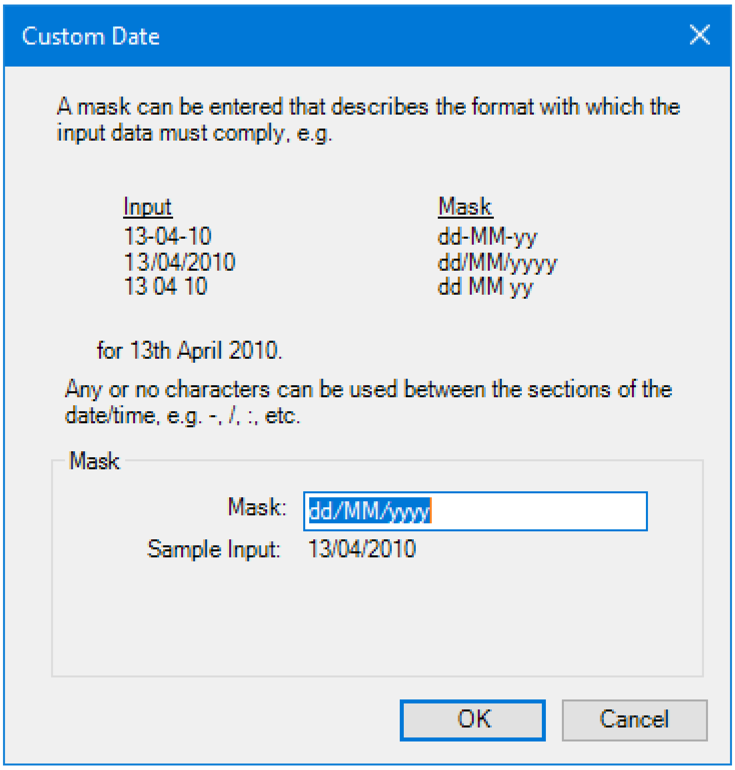
Without a header row in the input file, you must use **Add Column** to create the required recordset structure.

You must define the properties for each column:

▶ Column Properties

- □ **Name** — The name of the recordset column
- □ **Type** — The data type of the column. The default value is `string`.
- □ **Max Characters** — By default, the maximum number of characters for the column is `100`. Adjust this as required. Note that if a **Date**, **DateTime**, or **Time** data type has been selected then this property is not editable as the length is set automatically to match the selected format.
- □ **Format** — The data format of the recordset column. The default value is `Unformatted` in which case no validation takes place. If required, you can enter your own custom format.

If **Type** is either `Date`, `DateTime`, or `Time` and a `CUSTOM` format is selected, you are prompted to enter a **Mask** to describe the format, for example:



A mask can be entered that describes the format with which the input data must comply, e.g.

Input	Mask
13-04-10	dd-MM-yy
13/04/2010	dd/MM/yyyy
13 04 10	dd MM yy

for 13th April 2010.

Any or no characters can be used between the sections of the date/time, e.g. -, /, :, etc.

Mask

Mask: dd/MM/yyyy

Sample Input: 13/04/2010

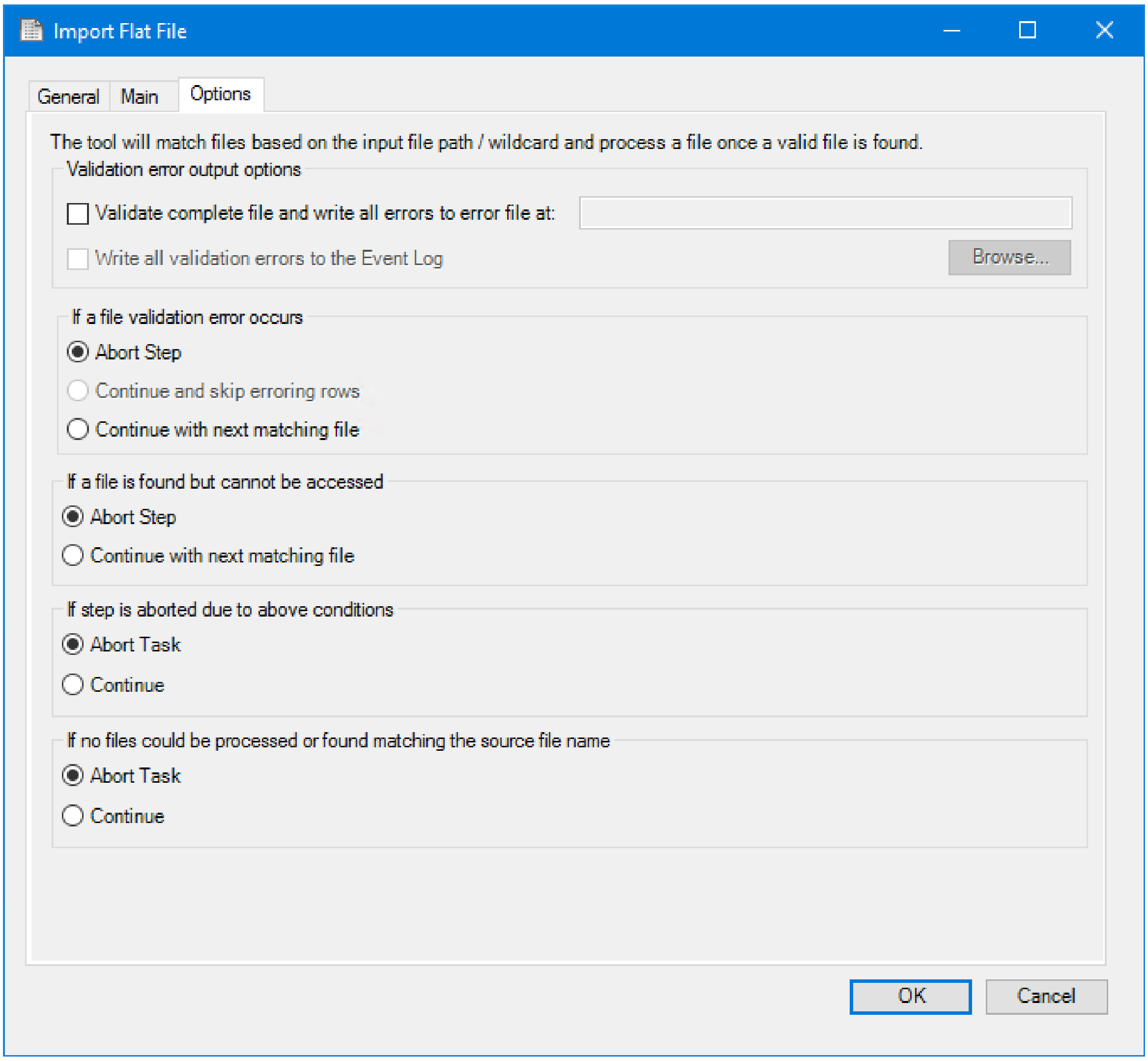
- □ **Required** — This property indicates whether a value is always required for this column at task run-time. Selecting **Yes** may cause the step to fail if a value is not present; **No** and the step continues even when the column is empty.
- □ **From** — If the input file is of fixed width format, this property states when the value for this column starts.
- □ **To** — If the input file is of fixed width format, this property states when the value for this column ends.
- □ **Validations** — Any data validation rules applied to this column — see Validating Data.

- **Preview Output** — Use this to check how the output is structured

- **Example input file contents** — If an example file has been defined in the File tab, this displays a small extract of that file as an aid to building your output

About the Options Tab

The **Options** tab allows you to define how errors in this step are handled at task runtime.



Import Flat File

General | Main | Options

The tool will match files based on the input file path / wildcard and process a file once a valid file is found.

Validation error output options

□ Validate complete file and write all errors to error file at:

□ Write all validation errors to the Event Log

Browse...

If a file validation error occurs

● Abort Step

○ Continue and skip erroring rows

○ Continue with next matching file

If a file is found but cannot be accessed

● Abort Step

○ Continue with next matching file

If step is aborted due to above conditions

● Abort Task

○ Continue

If no files could be processed or found matching the source file name

● Abort Task

○ Continue

OK | Cancel

If using data filtering validation rules (see Validating Data), you can choose to write errors to a location and file of your choice (**Validate complete file and write all errors to error file at**) — note that the folder location must exist at task run-time; the **Import Flat File** tool cannot create folders. Additionally, you can choose to **Write all validation errors to the Event Log** — note this is an "additional" option rather than an "or" option.

If a file validation error occurs you can choose to:

- **Abort Step** — Once an error is encountered in a file, the task moves onto the next step in the task sequence
- **Continue and skip erroring rows** — The whole input file is still processed and any errors in records encountered are skipped
- **Continue with next matching file** — Once an error is encountered in a file, the **Import Flat File** step moves onto the next input file awaiting processing, if any

If a file is found but cannot be accessed for whatever reason, you can choose to:

- **Abort Task** — Whatever the reason for not being able to access the input file, BPA Platform aborts the whole task
- **Continue with next matching file** — Whatever the reason for not being able to access the input file, the **Import Flat File** step moves onto the next input file awaiting processing, if any

If Step is aborted due to above conditions you can choose to :

- **Abort Task** — If the **Import Flat File** tool cannot access or validate any input files, then the whole task is aborted
- **Continue** — If the **Import Flat File** tool cannot access or validate any input files, then next step in the task sequence is started

If no files could be processed or found matching the source file name you can choose to:

- **Abort Task** — If no more input files can be found or all input files have validation problems, and there are no more input files present, the whole task is aborted
- **Continue** — The next step in the task sequence is started

Want to learn more?

Discover how Codeless Platforms can help your business by improving performance, boosting efficiency and cutting costs.

☎ +44 (0) 330 99 88 700

✉ enquiries@codelessplatforms.com

🧭 www.codelessplatforms.com

codeless
PLATFORMS