

FormXT - Form Extraction

Copyright © 2004-2007 Etasoft Inc.

Main website <http://www.etasoft.com>

FormXT website <http://www.xtranslator.com>

Purpose.....	2
Requirements	2
Licensing.....	2
Packaged Tools.....	2
Getting Started	2
Starting New Extraction.....	3
Adding New Fields	6
Scripts	13
Options	16
Command Line Processing	17
Software Integration via Developer SDK.....	18

Purpose

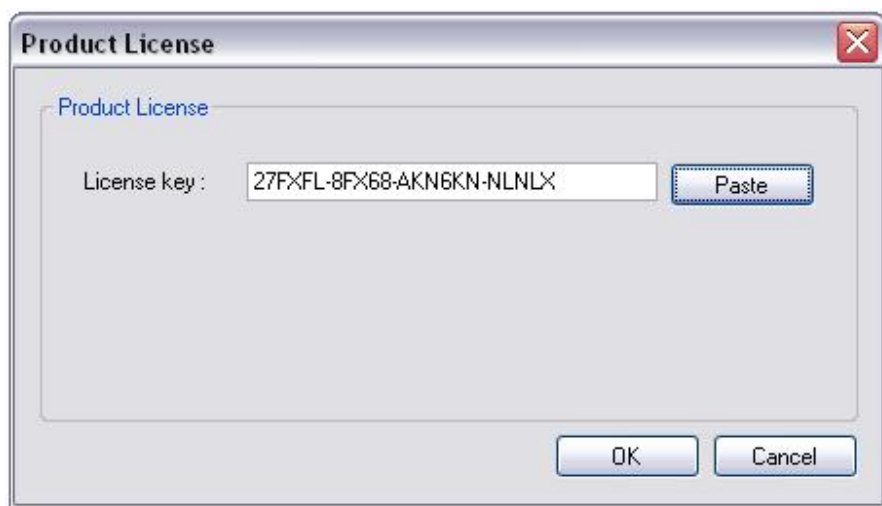
FormXT is a set of tools for data extraction from scanned or electronic forms and other type of flat text files. The backbone of it is extraction definitions you set using Extraction Editor. Once definitions are set you can save them into a file with extension *.fxt. You can use this file to run extractions using runner tools or integrate it into your software using Developer SDK for .NET. You can also automate the processing by using Extreme Processing product.

Requirements

<i>Minimum Requirements</i>	
Software	Windows 2000/XP/Vista/Server 2003
Hardware	Pentium 1GHz, RAM 256Mbt
<i>Recommended</i>	
Software	Windows XP Pro
Hardware	Pentium 1.5GHz, RAM 512Mbt

Licensing

Default package installation comes as time limited trial evaluation license. After fixed number of days starting from the day of the first use, trial evaluation license will expire and most of software features will become disabled. If you will like the product and purchase it, you will receive permanent license key. This new key must be entered in Extraction Editor License screen or passed as parameter "license=".



Extraction Editor License screen.

Packaged Tools

FormXT comes with number of tools. Tools are designed to fill three basic needs:

1. Extraction definition editing and testing.
2. Extraction execution using execution tools.
You can execute extraction using command line or Windows GUI based tool. Both tools provide simplest way to execute extractions in manual, semi-automated and automated environments.
3. Extraction execution via your program using Developer SDK for .NET.

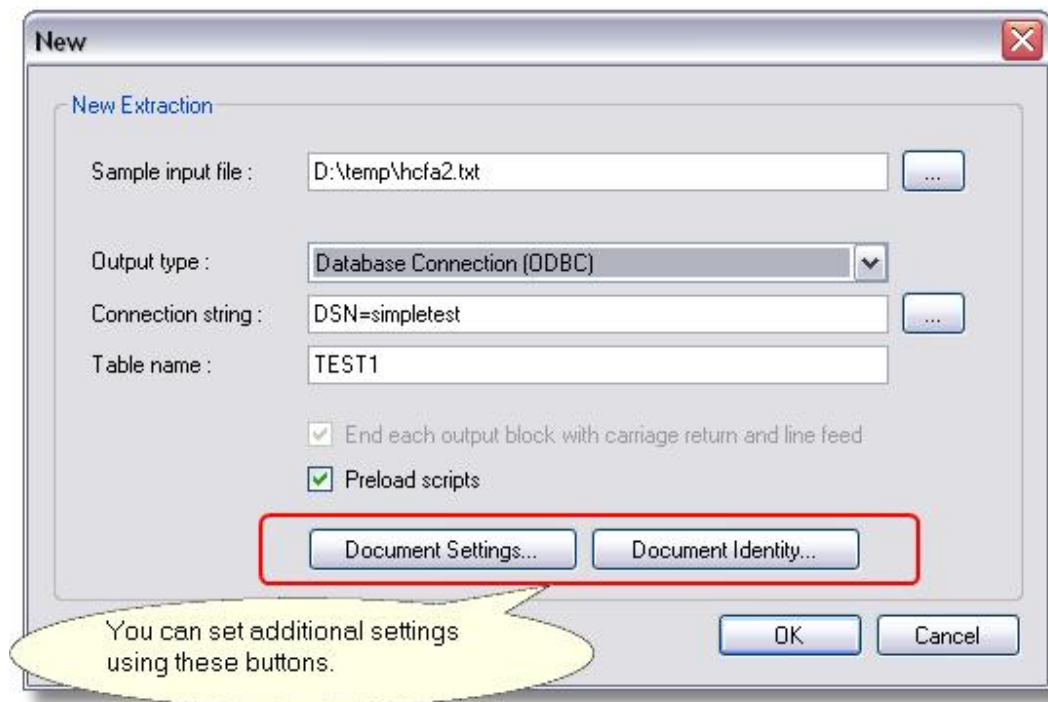
Getting Started

You will need sample file in order to setup fields to extract from it. Sample file should be actual production file you expect to receive once document is scanned and processed via scanner specific or OCR (Optical Character Recognition) software. Once you setup extraction fields for one sample file you can reuse these definitions for other files of the same type.

Often multiple scanned document forms are saved inside single file. FormXT can process them one by one using special options set in "Document Settings..." screen.

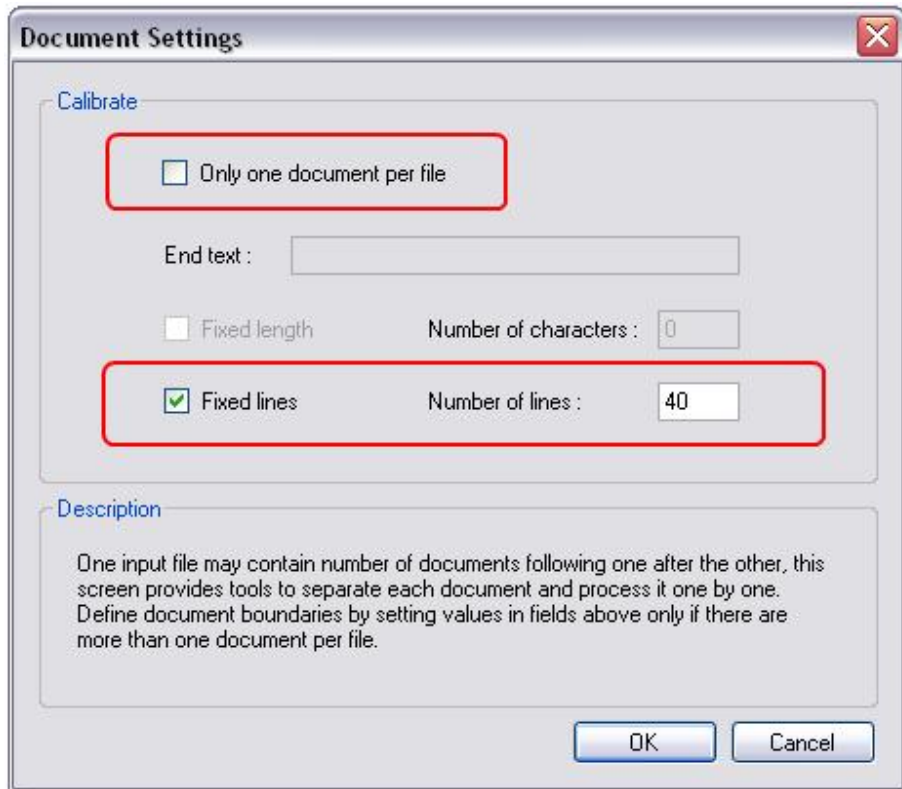
Starting New Extraction

New dialog screen will appear every time you start Extraction Editor. This same screen also appears when you press New menu or toolbar button. In this screen you can set number of settings for the new extraction such as input and output file names for testing, preload scripts, etc.



New dialog screen.

FormXT can extract data from files containing number of repeating documents. It can also identify documents in the file and skip over ones that are not identified. Use "Document Settings..." button if your input file may contain multiple of documents (not just one).



Document Settings dialog screen.

If there are number of documents inside single file, FormXT has to know when each document ends in order to separate them and process them one-by-one. Document Settings screen provides three different options to define document ending.

1. Document can end with specific text.
2. Document can have specific length in number of characters.
3. Document can have specific number of lines.

Third option is most often used and is the safest as scanned documents may have missing characters and first two options would fail to separate documents with scanning and OCR errors.

Identity

Identity

Check document for specific features

Identity 1 : 12345

AND (if unchecked then OR logic will be used)

Identity 2 :

Identity text has specific position

Start position : 1 End position : 20

Description

Input file may contain other data or junk, to identify the document you expect use tools provided on this screen. If identity properties are set on this screen but incoming document does not match them, it will not be skipped and not processed.

OK Cancel

Document Identity dialog screen.

If your file contains other documents and forms that are optional or you simply do not want to process them, use this screen to setup your document's identity. Identity is specific text or text combination in the document that identifies it uniquely. If identity is setup FormXT will automatically skip documents that do not match unique textual features of the expected document.

Form Extraction Editor - D:\temp\demo.fxt

File Edit View Project Help

Use these buttons to save extraction definitions.

Use these buttons to cut, copy, paste and add new fields.

Use this button to run and test extraction process.

TextView

X X X X X X
DOE, ILENE 11122222 X X DOE, JONE
6666 FRANCISCO LANE X X X X 6666 FRANCISCO L
SAN FRANCISCO MA X X X SAN FRANCISCO
99999 1234567890 X X X 55555 123456789
ABECK, ILENE 26782A
123456789A X X 122222222 X X
1111111 X X X X CA INTERNAL REVENUE SERV
INTERNAL REVENUE SERVICE X X YYYY ZZZZ
MEDICARE PART B CARRIERS RESERVED Y N
SIGNATURE ON FILE 12272003 SIGNATURE O
12345678 01012004 01012004 010120C
JANE A DUE MD A52456 01012004 01012C
RESERVED X X 123.00

216.5 216.5 MEDICAID ORIGINAL REF

Properties

Input field : PayerId

(Common)
Id 136
Name PayerId
Location End
EndsBefore

Output field :

(Common)
Id 137
Name PayerId_out
Basic
EndText
StartText
Script

Output [12] Errors / Warnings [0]

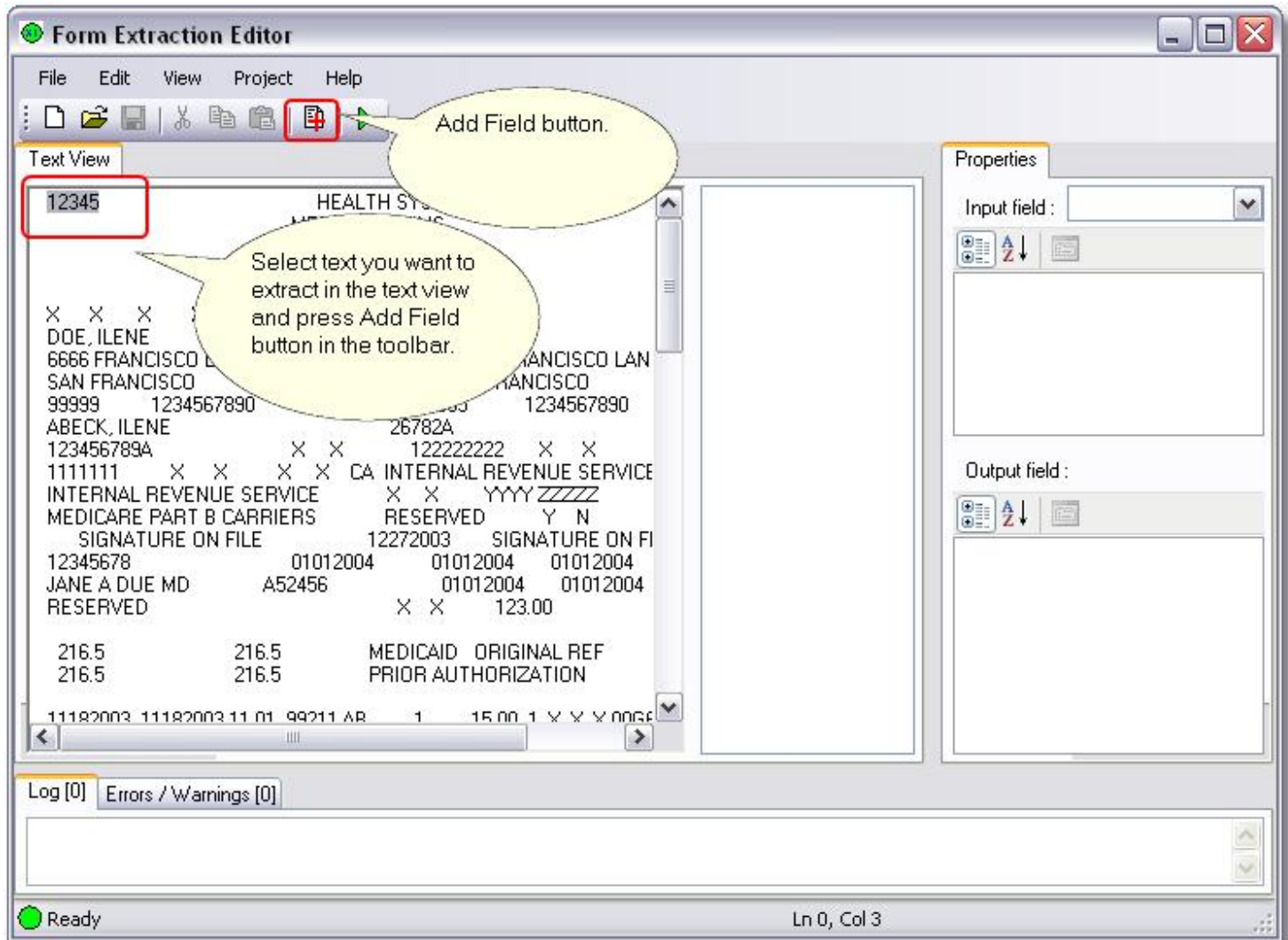
Transfer process finished
Output process finished
Completed

Processing finished successfully Ln 0, Col 3

This screen shows most important controls for working with extraction definitions and fields.

Adding New Fields

Once you load the sample file in TextView you can add fields by selecting the data and pressing Add Field button. New dialog will appear with settings for specific field.



Adding fields for extraction.

Add Field dialog screen.

There are two types of input fields: float and fixed. The difference between them is in how they starting position is defined. Floating fields must have other text that they will follow. Floating fields are used when data is not fixed inside the document by is prefixed with some constant text expression. For example: “Invoice No” or “Purchase Order Number” are usually words preceding specific invoice or purchase order numbers. If you are interested in actual number then floating field is the best way to extract it.

Some fields are best defined as being fixed. For example: fields in medical claim forms or tax forms are mostly fixed. If you selected text in the TextView tool will automatically use Line and Column of the selected text to be used as coordinates for the fixed field start.

Floating or fixed fields can have one of three possible endings defined. They can end with:

1. Specific text.
2. End with carriage return or line feed or space.
3. Have specific predefined length.

Extraction Editor automatically creates output field for you. It has the same name as input field but with “_out” added to the end of it. Output field provides extra formatting and validation options. You can attach your own scripts to perform formatting and validation or use already pre-built ones.

One typical issue with scanned documents is that they all have extra space characters surrounding the data you want to extract. In those scenarios fields are at the fixed position in the document but delimiters do not terminate they endings at the same time they can be of undefined length up to some maximum allowed length.

In those scenarios set FixedLength=True and set Length to maximum allowed length. Then use space trimming functions to remove excess of spaces. You can set property Function1 to TrimSpaces to perform space trim function.

Form Extraction Editor [Modified]

File Edit View Project Help

Text View

12345

Tool automatically created output field for you.

Additional input and output field properties you can change using these property tables.

Properties

Input field: PayerId

(Common)

Id 136

Name **PayerId**

Location End

EndsBefore

EndsWith **SpaceCRLF**

Output field:

(Common)

Id 137

Name **PayerId_out**

Basic

EndText

StartText

Script

FunctionScript

Log [0] Errors / Warnings [0]

Ready Ln 0, Col 3

Once field is added its properties are displayed on the right.

Form Extraction Editor [Modified]

File Edit View Project Help

Text View

12345 HEALTH SYSTEMS PayerId = PayerId_out

MEDICAL CLAIMS
PO BOX 55555
LEXINGTON, KY 55555

X X X X X X X 555888999
DOE, ILENE 11122222 X X D
6666 FRANCISCO LANE X X X X
SAN FRANCISCO MA X X X SA
99999 1234567890 X X X 55555
ABECK, ILENE 26782A
123456789A X X 122222222 X ;
1111111 X X X X CA INTERNAL REVENUE S
INTERNAL REVENUE SERVICE X X YYYY ZZZZ
MEDICARE PART B CARRIERS RESERVED Y N
SIGNATURE ON FILE 12272003 SIGNATUR
12345678 01012004 01012004 010
JANE A DUE MD A52456 01012004 010
RESERVED X X 123.00

216.5 216.5 MEDICAID ORIGINAL RE
216.5 216.5 PRIOR AUTHORIZATION

11182003 11182003 11 01 99211 AB 1 15.00 1 X X

Properties

Input field: PayerId

(Common)
Id 136
Name PayerId

Location End
EndsBefore
EndsWith SpaceCRLF

Output field:

(Common)
Id 137
Name PayerId_out

Basic
EndText
StartText

Script
FunctionScript

Log [0] Errors / Warnings [0]

Ready Ln 0, Col 50

Select next field to be extracted and press toolbar button Add Field again.

After first field has been added use same technique to add more fields.

Add Field

Input Field

Input field name : PayerName

Floating field Fixed field

Text before :

Line : 0

Column : 50

Fixed Length : 18 (pre-populated)

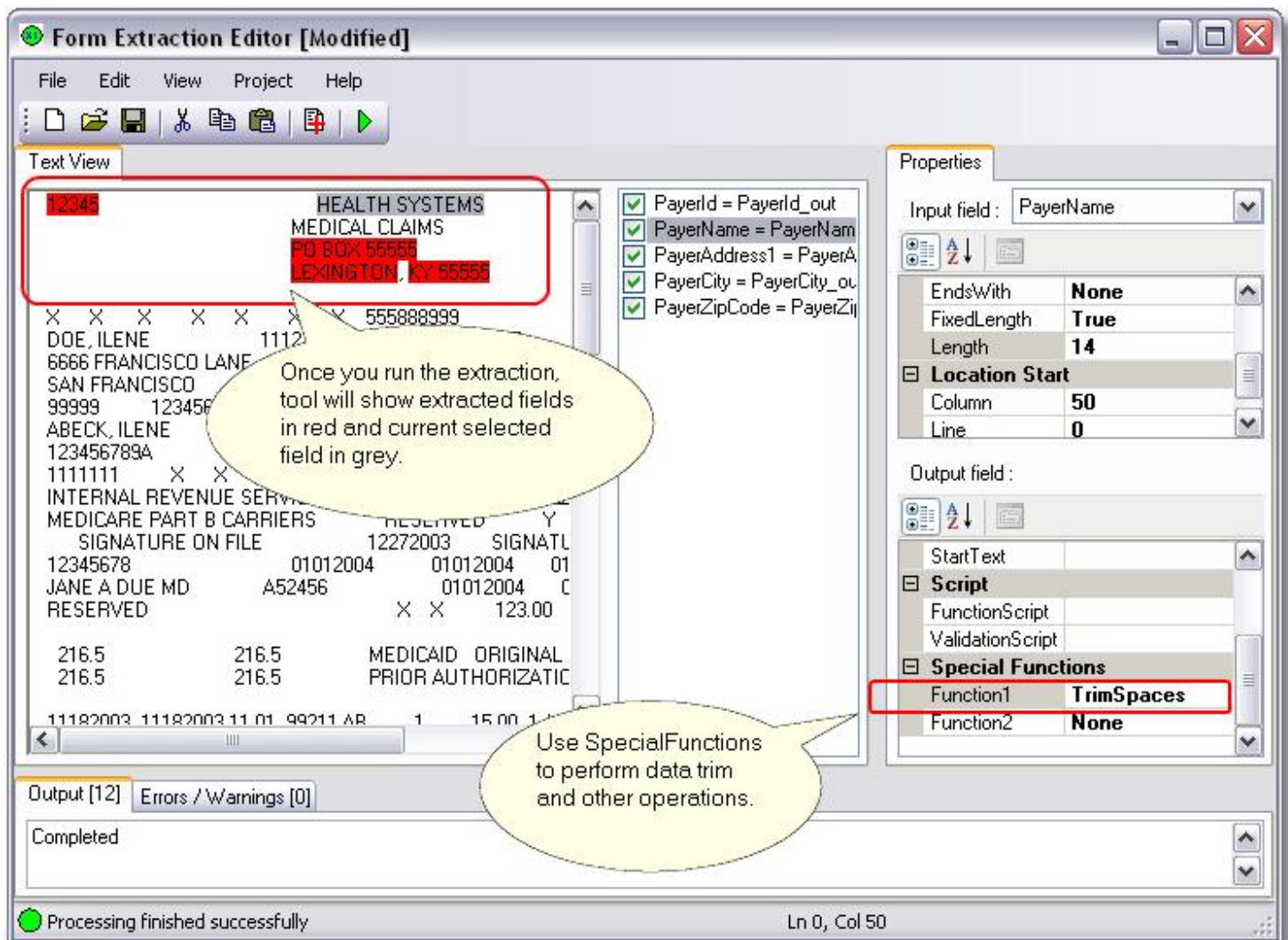
Text after : Use Selected Text

Special ending : None

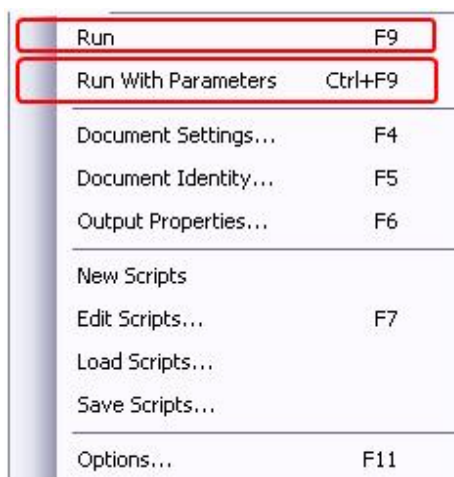
OK Cancel

We have changed pre-populated length to make sure that if Payer Name is longer it will not be truncated.

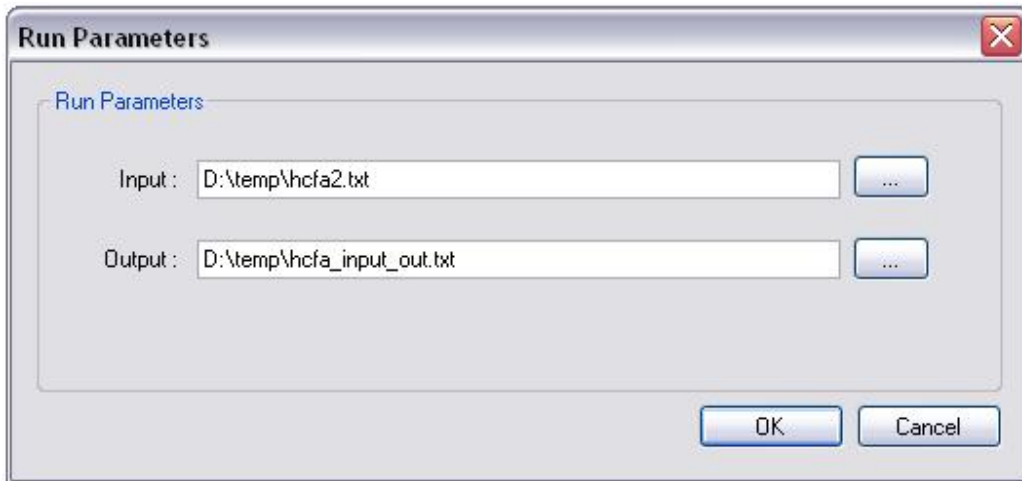
You can overwrite default values. For example: if you expect text to be longer on some documents than this sample you use now, change length to the value that is more accurate and use functions to truncate excess spaces.



You can change almost every property later by using right panel.



Pressing one of these menu options will run extraction definitions and update TextView with extracted fields colored in red.

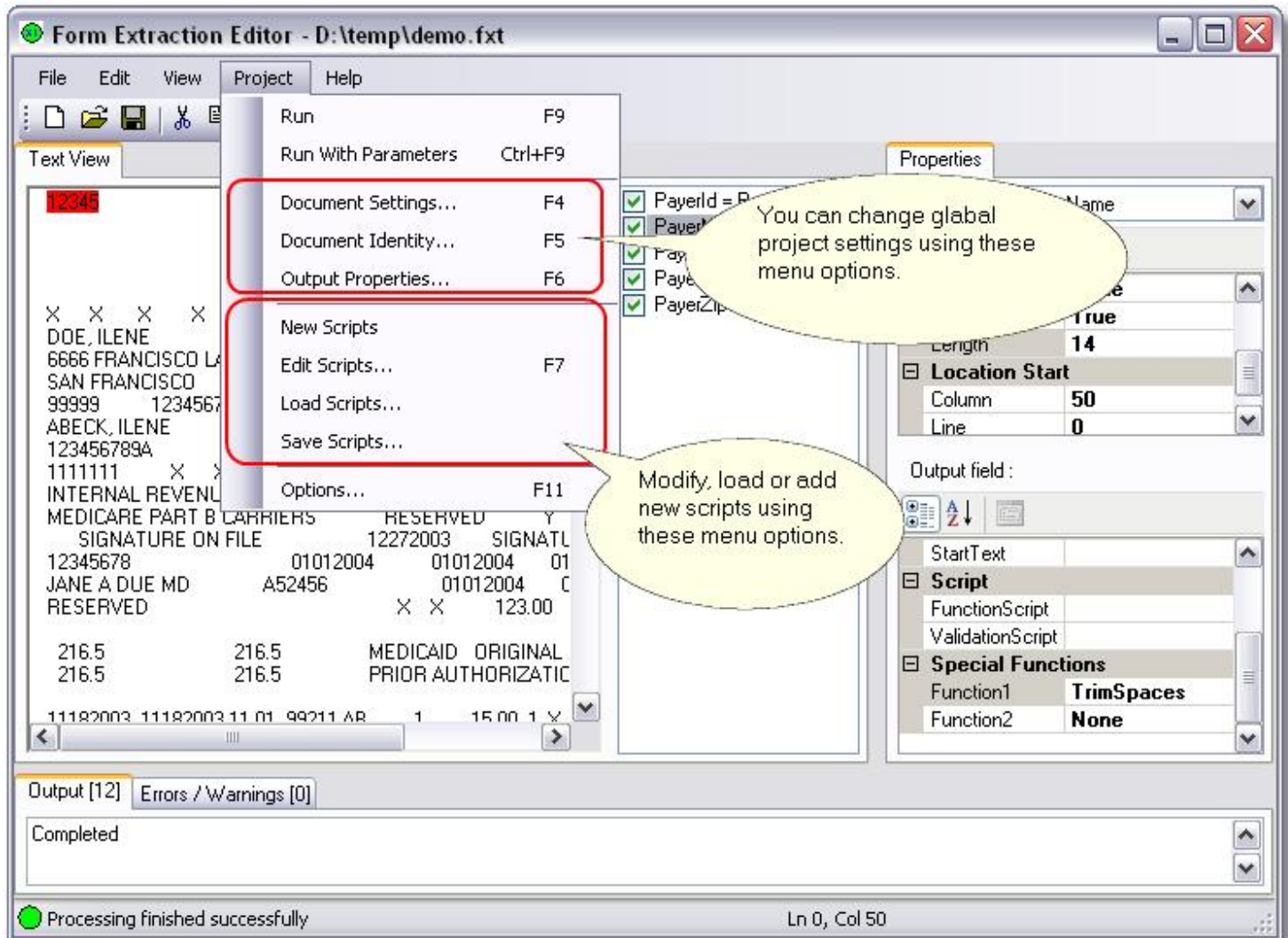


You can use menu option "Run With Parameters" in order to try other sample files against fields you already defined.

Scripts

Scripts are small formatting or validation functions you can attach to the field. During processing once field is extracted and placed into the output, functions and scripts attached to it will be executed.

Some pre-built example scripts come in the file "scripts.src". You can load those scripts using Project->Load Scripts menu option. They will also load if you mark a checkbox "Preload scripts" in New dialog screen.



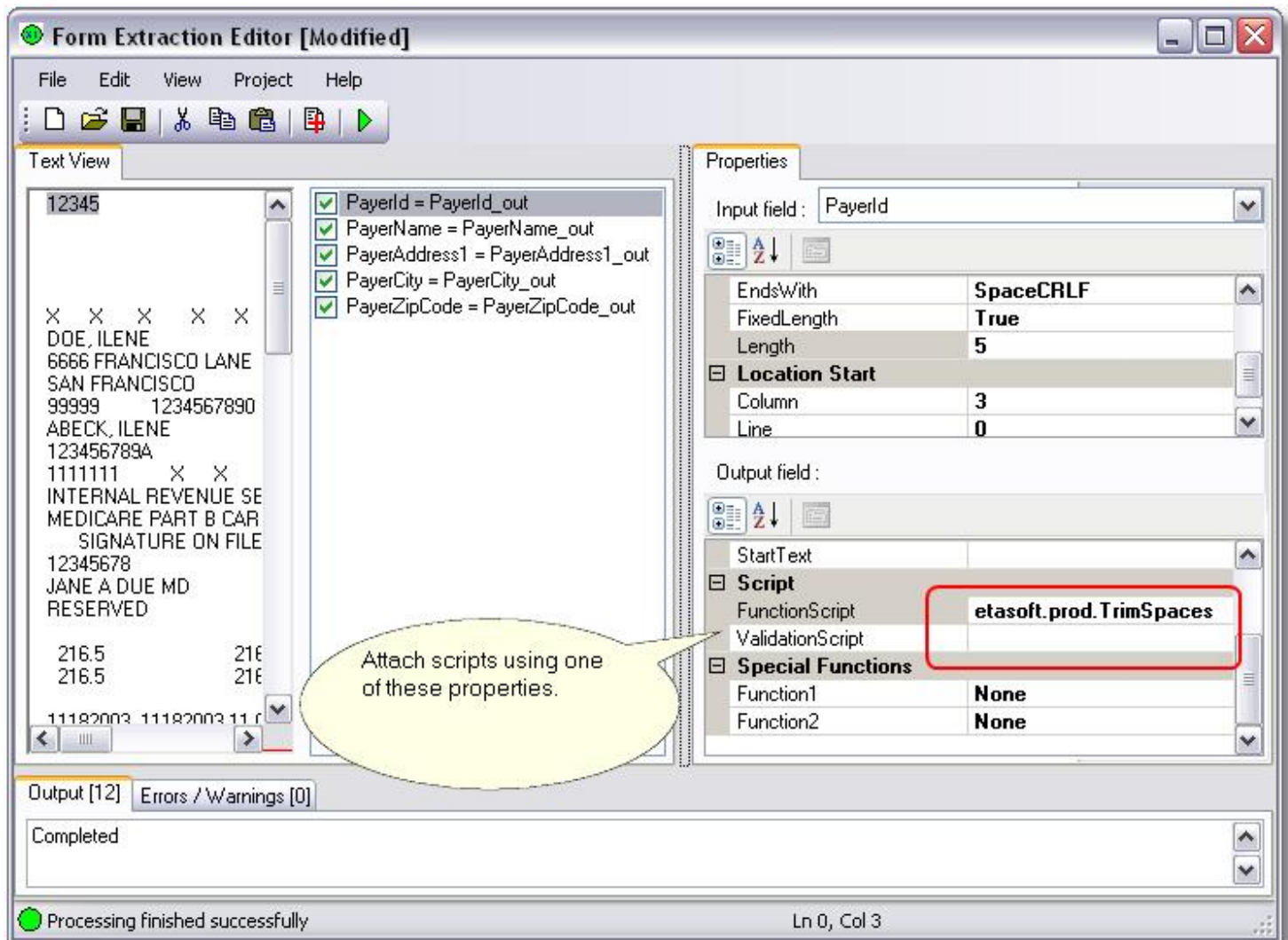
You can save existing scripts into a file and reuse them when you define other extractions.



You can write scripts in C#.NET. There is simple script that just trims spaces from incoming data.

Id variable in the script contains Id property value of the field that is executing this script. You can attach the same script to more than one field. You can use Id property to identify specific field.

strData variable contains actual input data that is being extracted. For formatting functions if you modify it and use return statement to return it, modified data will be placed into the output. For validation functions if you want validation to fail with validation specific error, return null any other returned string value will not result in validation error being reported.



You can attach scripts to FunctionScript and ValidationScript properties.

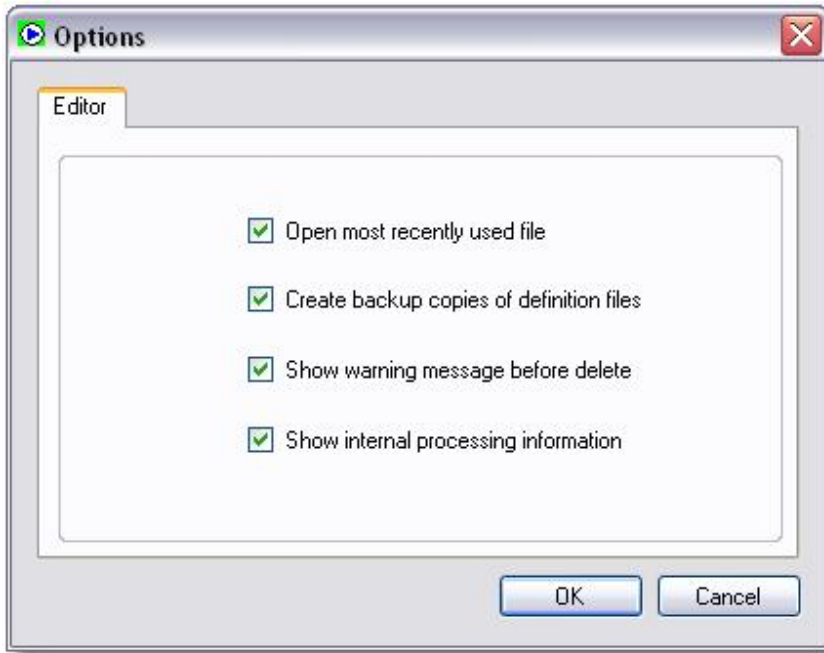
Result string returned from FunctionScript will replace whatever data was coming from the input. If you simply want to return the same data that was extracted use "return strData" in your script and if strData was not modified, it will result into no changes to the output data.

If script attached to ValidationScript will return null, it will automatically result into validation error. So if you want to validate data and within the script you find out something wrong with the incoming data, return null and validation error will be reported.

Options

Extraction Editor has some basic options accessible via Options menu:

1. "Open most recently used file" this option if checked will open last used *.fxt extraction definition file on Extraction Editor startup.
2. "Create backup copies for definition files" will create backup copies of *.fxt files on every save.
3. "Show warnings message before delete" will display warning message when extraction field is being deleted or cut.
4. "Show internal processing information" is useful when unexpected errors occur during extraction execution. If this option is turned on extra processing information will be produced in "Log" tab.



Options dialog screen.

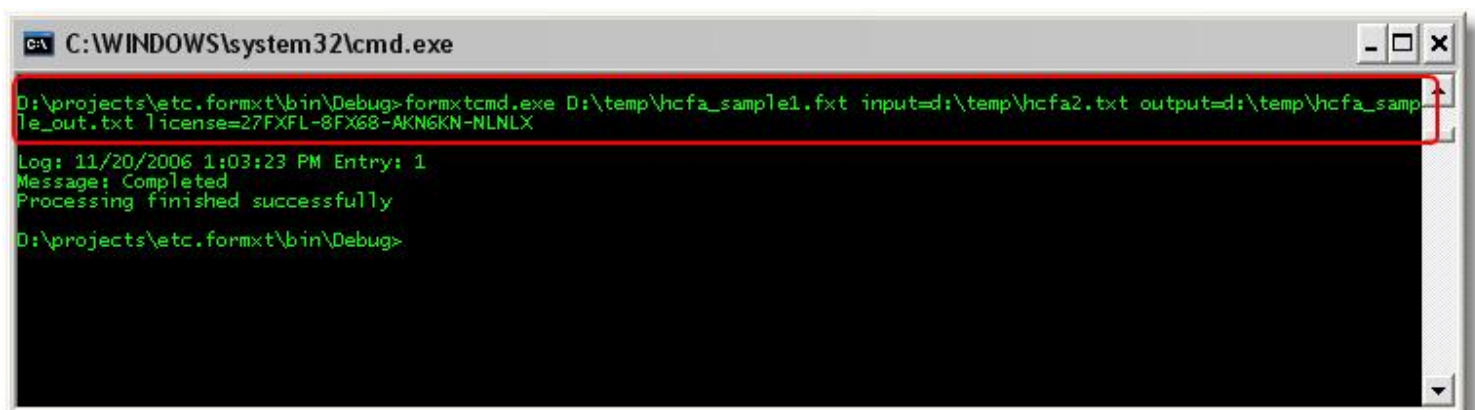
Command Line Processing

Once you have saved extraction definitions via Extraction Editor you can run them using execution tools or integrate them into your programs using Developer SDK. One of the tools that come with the package is command line (DOS shell based) execution tool called "formxtcmd.exe". This tool needs 4 parameters to run:

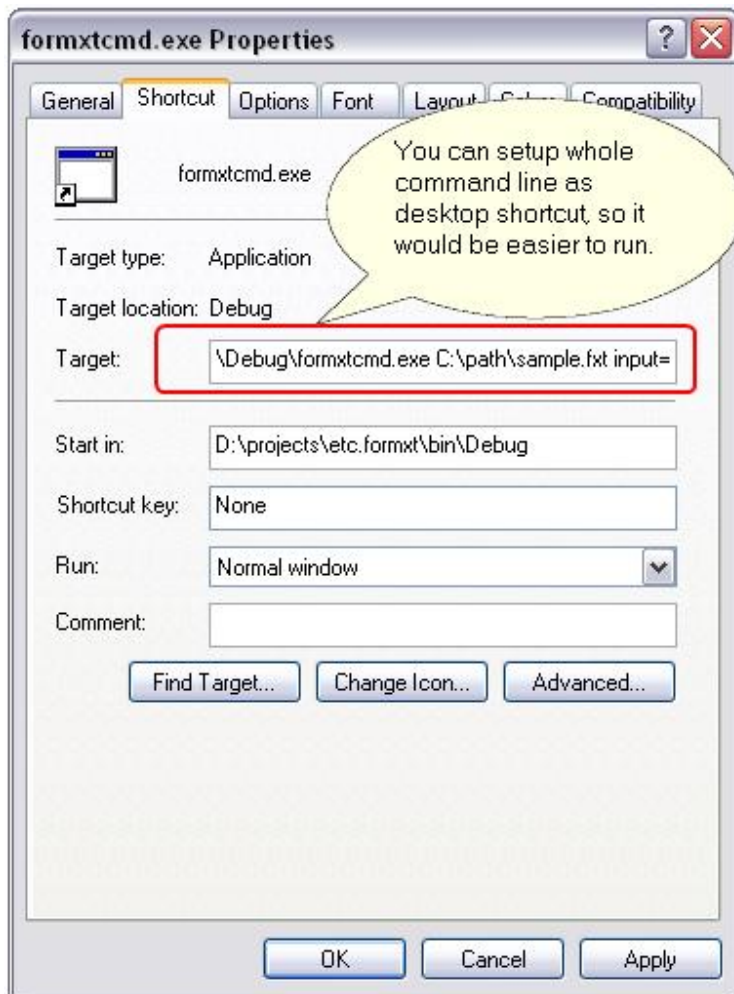
1. File with extraction definitions (file with extension *.fxt).
2. Input file name with path to it.
3. Output file name with path to it.
4. License key from Extraction Editor "Product License" screen (accessible via menu Help->Product License).

Whole command line may look like:

```
formxtcmd.exe C:\path\sample.fxt input=C:\path\input.txt output=C:\path\output.txt license=YYYYZZXXX
```



You can also setup shortcut on the desktop so it would be easier to run than typing and retying it all in the command line.



Desktop shortcut screen.

Software Integration via Developer SDK

DeveloperSDK folder contains C#.NET source code for direct FormXT processing engine integration into your software. Basic functions available via Developer SDK: load extraction definitions file *.fxt, run extraction, handle errors. C#.NET source code is actual source code for formxtcmd.exe utility (command line extraction runner).