

# eFORMz Mini-Manual

## Text to XML Converter: QAD

---

---

# Minisoft® eFORMz™

---

---

Version 10.0

Minisoft, Inc.  
1024 First Street  
Snohomish, WA 98290  
U.S.A.

1-800-682-0200  
360-568-6602  
Fax: 360-568-2923

Minisoft Marketing AG  
Papiermühleweg 1  
Postfach 107  
Ch-6048 Horw  
Switzerland

+41-41-340 23 20  
info@minisoft.ch  
www.minisoft.ch

Internet access:

sales@minisoft.com  
support@minisoft.com  
<http://www.minisoft.com>  
<http://www.minisoft.us>

# Disclaimer

The information contained in this document is subject to change without notice.

Minisoft, Inc. makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Minisoft, Inc. or its agents shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishings, performance, or use of this material.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another programming language without the prior written consent of Minisoft, Inc.

©2014 by Minisoft, Inc. Printed in U.S.A.

eFORMz 10.0 - January 2014

All product names and services identified in this document are trademarks or registered trademarks of their respective companies and are used throughout this document in editorial fashion only and are not intended to convey an endorsement or other affiliation with Minisoft, Inc.

# License Agreement

READ CAREFULLY BEFORE INSTALLING THE MINISOFT SOFTWARE APPLICATION:

CUSTOMER: THE MINISOFT SOFTWARE APPLICATION ("PRODUCT") THAT YOU PURCHASED CONTAINS COPYRIGHTS, TRADE SECRETS, TRADE MARKS, AND OTHER INTELLECTUAL PROPERTY RIGHTS BELONGING TO MINISOFT, INC. ("MINISOFT"), HAVING ITS PRINCIPAL OFFICES LOCATED AT 1024 FIRST STREET, SNOHOMISH, WASHINGTON, 98290, U.S.A. THE PRODUCT MAY BE INSTALLED, COPIED, DISPLAYED, EXECUTED, PERFORMED, OR USED IN ANY OTHER FASHION ONLY IN ACCORDANCE WITH THE TERMS OF THIS LICENSE AGREEMENT ("AGREEMENT" OR "LICENSE"). THE TERMS "YOU" OR "YOURS" ARE SOMETIMES USED HEREIN AND MEAN RESPECTIVELY CUSTOMER OR CUSTOMER'S.

SOMETIMES EACH OF MINISOFT AND CUSTOMER IS REFERRED TO HEREIN AS A "PARTY," AND SOMETIMES THEY ARE COLLECTIVELY REFERRED TO AS THE "PARTIES."

YOU MAY NOT INSTALL, COPY, DISPLAY, EXECUTE, PERFORM OR USE IN ANY OTHER FASHION THE PRODUCT EXCEPT IN ACCORDANCE WITH THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU INSTALL, COPY, DISPLAY, EXECUTE, PERFORM OR OTHERWISE USE ANY PART OF THE PRODUCT, YOU ARE DEEMED TO HAVE ACCEPTED AND YOU ARE BOUND BY ALL THE TERMS AND CONDITIONS OF THIS MINISOFT LICENSE AGREEMENT ("AGREEMENT"). IN ORDER TO USE THE PRODUCT, YOU MAY BE REQUIRED TO OBTAIN A UNIQUE CODE OR CODES FROM MINISOFT TO ENABLE INSTALLATION, COPYING, DISPLAY, EXECUTION, PERFORMANCE AND OTHER USE OF THE SOFTWARE PRODUCT.

IF YOU DO NOT ACCEPT AND AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT, YOU ARE PROHIBITED FROM INSTALLING, COPYING, DISPLAYING, EXECUTING, PERFORMING, OR USING IN ANY OTHER FASHION THE PRODUCT. VIOLATION OF THIS AGREEMENT (1) CONSTITUTES INFRINGEMENT, MISAPPROPRIATION AND CONVERSION OF MINISOFT'S COPYRIGHTS, TRADE SECRETS, AND OTHER INTELLECTUAL PROPERTY RIGHTS, (2) WILL SUBJECT THE INFRINGER TO SIGNIFICANT CIVIL LIABILITIES FOR DAMAGES (INCLUDING WITHOUT LIMITATION STATUTORY DAMAGES) AND OTHER REMEDIES AND LIABILITIES, AND (3) MAY CONSTITUTE VIOLATION OF CRIMINAL LAWS.

IF A CUSTOMER IS PURCHASING AN UPGRADE FOR AN EARLIER VERSION OF THE PRODUCT, THEN THIS AGREEMENT SHALL SUPERSEDE THE TERMS AND CONDITIONS OF THE LICENSE AGREEMENT THAT APPLIED TO THE EARLIER VERSION AS TO THE USE OF THE UPGRADE PRODUCT AS OF THE DATE OF INSTALLATION OF THE UPGRADE.

NOW, THEREFORE, in consideration of the foregoing provisions and the terms and conditions herein contained and for other good and valuable consideration, the receipt and sufficiency of which are hereby acknowledged, the following terms and conditions apply:

## A. License Grants

- Upon MINISOFT's receipt from the Customer of the one-time license fee and further subject to the other terms and conditions herein, MINISOFT grants the Customer a non-exclusive and non-transferable perpetual license: (a) to install and use the Product on only one host computer or business server in the Customer's business organization; and (b) to install the client portion of the Product on a network server for access by more than one personal computer and on any personal computer client accessing the host server program.
- You may not: (a) allow access to or use of the Product to anyone outside of your organization; (b) decompile, reverse engineer, decrypt, disassemble, or otherwise attempt to copy or replicate the source code; (c) modify or translate those portions of the Product that are provided in an object or source code form, or alter, merge, modify, reverse engineer or adapt the Product in any way including disassembling or decompiling; (d) copy, loan, rent, lease or sublicense, assign, convey, sell, mortgage, pledge, dispose or in any other manner transfer to a third-party any part of or all of the Customer's rights and obligations under this Agreement or in the Product; or (e) otherwise copy, distribute, rent, loan, disseminate, publish, sell or transfer any portion of the Product in any way inconsistent with or contrary to the terms and conditions of the Agreement.
- Any attempt to violate any prohibitions described in this Agreement shall be considered a material breach of this Agreement, and no attempt to violate section 2(c) or 2(d) in the immediately preceding paragraph shall have any effect, and any prohibited act described therein shall be void and of effect if made by the Customer.
- You may make one copy of the Product for use as a backup copy (the "Copy") provided that: (i) the Copy includes all notices of copyright and all other proprietary rights appearing in and on the Software Product; (ii) the Copy and the original Product are not in use at the SAME time; (iii) Customer establishes a procedure for accounting for the Copy at all times; and (iv) Customer destroys or returns to MINISOFT the Copy when it is no longer required or when the license grants made hereunder are terminated for any reason, whichever occurs first.
- This Agreement grants to the Customer a license for use of the Product on the terms and conditions set forth in this Agreement; and (ii) the Customer may not install, copy, perform, execute, display, or use in any other fashion the Product except in accordance with the terms and conditions of this Agreement. By any use of any element of the Product, the Customer shall be deemed to have accepted and be bound by all the terms and conditions of this Agreement.
- MINISOFT retains and reserves all rights to the Product. Nothing contained herein shall be deemed to convey to the Customer any title to, or an ownership interest in, the Product or any of the intellectual property rights contained in the Product. Upon termination of this Agreement, all rights granted to the Customer herein shall automatically revert to MINISOFT.

## B. Termination

The license grants made hereunder shall automatically terminate on the date of any breach by the Customer of any of the terms and conditions of this Agreement. In any such termination, you shall discontinue using the Product. You further agree to return the Product to MINISOFT or destroy all components of the Product and to certify in writing that the Product is no longer in use and that you have kept no copies of the Product. In any such termination of the license grants, none of MINISOFT's rights in law or equity shall be barred, discharged, released, or otherwise impaired by such termination, and MINISOFT reserves all such rights described herein.

## C. Confidential Information

The Product also contains MINISOFT's commercially valuable, proprietary and confidential information and trade secrets (collectively "Confidential Information"). You are obligated to preserve and hold in confidence any Confidential Information and may not use any of it except in accordance with the terms and conditions of this Agreement. Confidential Information shall not include what is already public at the date of its disclosure or what later becomes lawfully public through no fault or responsibility of the Customer.

## D. Limited Warranties and Limitations on Damages and Remedies.

- Only the limited warranties expressly described in this section are made to the Customer purchasing this Product. No warranty of any kind, whether express or implied, is made to any third-party or other transferee or successor of the Customer.
- MINISOFT warrants that it has the right to grant the licenses contained in this Agreement.
- MINISOFT warrants that this product will execute its programming instructions when properly installed on a properly configured computer for which it is intended. MINISOFT makes no warranty of any kind, implied or express, that the Product will meet Customer's requirements or that the Product will be uninterrupted or error-free.
- MINISOFT, Inc. warrants the media upon which this Product is recorded to be free from defects in materials and workmanship under normal use for a period of 90 days from the date of purchase. During the warranty period MINISOFT will replace media that prove to be defective. Customer's exclusive remedy for any media that proves to be defective shall be to return the media to MINISOFT for replacement.
- MINISOFT's only obligation, liability and/or remedy, with respect to the warranties set forth in this Agreement shall be one of the following, which MINISOFT shall have sole discretion to choose: (1) as described above, to provide replacement media (whether in the form of the current release or otherwise) for the Product; (2) to replace, without charge, the Product with a functionally equivalent software product; or (3) to refund the applicable license fees paid to MINISOFT by Customer.

These are the sole and exclusive remedies for any breach of warranty.

6. MINISOFT shall have no obligation, liability or any other kind of responsibility for the costs of restoring any destroyed or defective data, installation of replacement media, or software conversion work made necessary by replacement media, even if MINISOFT had been specifically informed by Customer of the possibility or probability of any such consequences. During the ninety (90) days warranty period for the media, MINISOFT will replace, without charge, on an exchange basis, any media which is not as warranted. Customer must return the defective item postpaid to MINISOFT, postmarked within the time period stated above. Customer must either insure the defective item being returned or assume the risk of loss or damage in transit. Any claim under the above warranty must include a copy of Customer's receipt or invoice or other proof of the date of delivery.

7. Year 2000 Warranty. MINISOFT warrants that the Product will accurately process date/time data (including, but not limited to, calculating, comparing, and sequencing) from, into, and between the twentieth and twenty-first centuries, and the years 1999 and 2000 and leap year calculations, in accordance with the Product's documentation BUT ONLY to the extent that other technology and systems (e.g., software and operating systems, hardware and firmware) used in combination with the Product properly exchanges date/time data with it. In addition to the limitations of liability and remedies described above in this Section, the Year 2000 warranty made herein is made only for ninety (90) days after the date of delivery of the Product and only if any noncompliance is discovered and made known to MINISOFT in writing within such ninety (90) days. The Year 2000 warranty expires after such ninety (90) day period and shall not extend to any events occurring after the expiration date.

8. MINISOFT makes no other warranty, expressed or implied, with respect to the Product.

9. NOTICE. THE WARRANTIES, OBLIGATIONS AND LIABILITY OF MINISOFT AND THE RIGHTS AND REMEDIES OF CUSTOMER SPECIFICALLY SET FORTH IN THIS AGREEMENT ARE THE CUSTOMER'S EXCLUSIVE REMEDIES. MINISOFT DISCLAIMS, AND CUSTOMER HEREBY WAIVES, DISCHARGES AND RELEASES MINISOFT, ITS OFFICERS, DIRECTORS, EMPLOYEES AND AGENTS, FROM ANY AND ALL (1) OTHER WARRANTIES, OBLIGATIONS, OR LIABILITIES AND (2) RIGHTS, CLAIMS, CAUSES OF ACTIONS, EXPRESS OR IMPLIED, ARISING BY LAW OR OTHERWISE IN CONNECTION WITH, REGARDING, OR ARISING FROM THE PRODUCT OR ANY RELATED SOFTWARE, TANGIBLE OR INTANGIBLE ITEMS OR SERVICES PROVIDED UNDER THE AGREEMENT (INCLUDING, WITHOUT LIMITATION, ALL DOCUMENTATION SUPPLIED WITH THE PRODUCT). THE WAIVER, DISCHARGE AND RELEASE INCLUDE, BUT ARE NOT LIMITED TO: (1) ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A SPECIFIC PURPOSE; (2) ANY IMPLIED WARRANTY ARISING FROM COURSE OF PERFORMANCE, COURSE OF DEALING, OR USAGE OF TRADE; AND (3) ANY OBLIGATION, LIABILITY, RIGHT, CLAIM, OR REMEDY, WHETHER IN TORT, CONTRACT, OR OTHERWISE. IN NO CASE SHALL MINISOFT BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, LOSS OF USE, REVENUE, PROFIT, OR DATA), PUNITIVE DAMAGES, EXEMPLARY OR OTHER STATUTORY DAMAGES, EXPENSES, COSTS (INCLUDING ATTORNEY FEES). IN ADDITION, IN NO CASE SHALL MINISOFT'S LIABILITY EXCEED THE FEES PAID BY CUSTOMER FOR THE PRODUCT.

#### E. Updates

Customer is entitled to updates and enhancements to licensed software if under a current Minisoft software update and subscription service. The cost of the update and subscription service will be 20% of the purchase price of the licensed software. The update and subscription service is for twelve (12) months and can be renewed at the customer's discretion.

#### F. Support Services

A customer that has purchased annual support entitles them to any and all software updates for their licensed product(s) as well as technical support via phone and email. Software Updates are available from our web site, [www.minisoft.com](http://www.minisoft.com). Included with the purchase of support is the use of our Toll Free 800 number, and support email. Days and Hours of support coverage are Monday through Friday 7:30am (Pacific) until 5:00pm (Pacific). When support is purchased or renewed, an update may be downloaded from the Minisoft Website. In cases where support has expired, back-support is charged to bring the support account current. Any supplemental software code provided to you as part of the Support Services shall be considered part of the SOFTWARE PRODUCT and subject to the terms and conditions of this License Agreement.

#### G. Miscellaneous

1. Governing Law. This Agreement is made for the sole benefit of the parties hereto, and there are no third party beneficiaries to this Agreement. This Agreement shall be interpreted under the laws of the State of Washington, notwithstanding the application of any jurisdiction's choice-of-law rules to the contrary.

2. Independent Contractors. No agency, partnership, joint venture, or employer-employee relationship is created hereby, and neither party nor any of its agents or employees shall have any authority of any kind to bind the other party in any respect whatsoever. Each party is an independent contractor to the other party.

3. No Assignment by Customer. This Agreement shall be binding upon and inure to the benefit of MINISOFT and any successors and assigns. Customer shall have no right to assign or otherwise transfer its rights or obligations under this Agreement except with the prior written consent of MINISOFT, which MINISOFT in its sole discretion and for any reason or no reason provide or withhold.

4. Entire Agreement. This Agreement contains the complete understanding between the parties and shall as of the date the Product is first installed supersede all other agreements, whether they are written or oral, between the parties concerning the particular subject matter. The language of this Agreement shall for all purposes be construed as a whole, according to its fair meaning, not strictly for or against either party, and without regard to the identity or status of any person who drafted all or any part of it. No purchase order or any other purchasing instruments issued by Customer, even if such purchase order or other purchasing instrument provides that it takes precedence over any other agreement between the parties, shall be effective to contradict, modify, delete from or add to the terms of this agreement in any manner whatsoever.

5. Severability. In the event that any one or more provisions of this Agreement is found by a court of competent jurisdiction to be unenforceable or invalid, then notwithstanding any such finding the remainder of this Agreement shall remain in full force and effect, and such provision that is found to be unenforceable or invalid shall be deemed severed.

6. Waiver. Neither party's right to require performance of the other party's obligations hereunder shall be affected by any previous waiver, forbearance, or course of dealing, nor shall any waiver or forbearance or other course of dealing at any time with respect to any term or condition in this Agreement be effective unless evidenced in writing signed by the waiving party as to such waiver, forbearance or other course of dealing. No waiver or modification of this Agreement or any covenant, condition or limitation herein contained shall be valid and no evidence of waiver or modification shall be offered or received in evidence in any proceeding, arbitration or litigation between the parties hereto arising out of or affecting this Agreement or the rights or obligations of the parties hereunder, unless such waiver or modification is in writing duly signed by both parties.

7. Headings, Sections - The various headings in this Agreement are inserted for convenience only and shall not affect the meaning or interpretation of this Agreement or any Section or provision hereof.

8. Disputes; jurisdiction and venue. Customer and MINISOFT consent to personal jurisdiction, subject matter jurisdiction, and venue in the state and/or federal courts sitting in King or Snohomish Counties, State of Washington. Any action relating to this Agreement must be brought in the state or federal courts located in King or Snohomish County, Washington.

9. Attorney's Fees. In the event legal action is brought by either party to enforce any of the provisions of this Agreement, the prevailing party shall recover its reasonable attorney's fees, costs, and expenses, including but not limited to fees, costs and expenses of collecting any judgment.

10. Export. The export of this product is restricted by U.S. export regulations.

11. GOVERNMENT RESTRICTED RIGHTS. The Software Product and documentation are provided with restricted rights. Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (2) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252. 227-7013 or subparagraphs (c) (1) and (2) of the Commercial Computer Software-Restricted Rights at 48 CFR 52.227-19, as applicable. Contractor/manufacturer is MINISOFT, INC., 1024 FIRST STREET, SNOHOMISH, WASHINGTON, 98290, U.S.A.

## Table of Contents

# eFORMz Mini-Manual - Text to XML Converter: QAD

## Text to XML Conversion

Overview .....	10
<b>What does it do?</b> .....	<b>10</b>
<b>How does it work?</b> .....	<b>10</b>
<b>When should the preprocessor be used?</b> .....	<b>12</b>
<b>When should the preprocessor not be used?</b> .....	<b>12</b>

## Configuring Text to XML

Setting a Preprocessor for Text to XML Conversion .....	14
--	----

## The Text to XML Editor

Line and State Viewer .....	21
Adding Lines & Parameters in the Text to XML Editor .....	22
Adding States in the Text to XML Editor .....	25

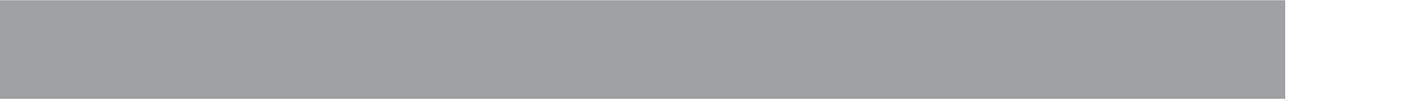
## Configuring for QAD

Text to XML Editor	
Adding Lines & Parameters .....	35
Troubleshooting with	
Lines & Parameters .....	38
Comments & Remarks .....	38
Detail Lines .....	43
White Space .....	50
Adding States in the Text to XML Editor .....	51
Troubleshooting with States .....	54

## Reference Guide

State Engine .....	62
Text to XML Editor Window .....	63
Line Editor Window and Options.....	64
Line Editor Parameter Options.....	65
State Editor Window.....	66
<b>State Editor Options</b> .....	<b>66</b>
<b>EndState Options</b> .....	<b>67</b>
The Input Data Viewer .....	68
The XML Data Viewer .....	69

## Glossary



# *Text to XML Conversion*

---

# Overview

Creating an XML configuration file has many advantages for the eFORMz user. By using the Text to XML preprocessing feature, text can be “parsed” into XML for subsequent processing. This is especially useful when source data spans multiple pages. After conversion, the page structure is left behind so that the final output can include more or less pages than the source. To do this, the eFORMz user must be familiar with the Text to XML Converter and the Text to XML Editor.

## What does it do?

The transformation of Text to XML is the process by which information that was originally intended for pre-printed forms is converted into an annotated and tabular format, prior to enhancement by a Form Template.

The annotations ensure a more user-friendly data selection environment. Each of the items from the original Print Output will have a meaningful name.

Conversion to a tabular format also allows for automating the process of reformatting repeating data without regard to the source placement on a page. If the original Print Output was multiple pages, the tabular arrangement (XML) will permit a meaningful repagination process (*See Samples 1 and 2*).

## How does it work?

The Text to XML conversion occurs when the Text to XML preprocessor is configured for a Form Template. The configuration includes a set of rules and definitions for locating, extracting and naming data elements from the original Print Output.

It uses a collection of Lines, Parameters, States and EndStates to define how the source data is parsed into the XML output. (See References – An Explanation of the State Engine)

**Sample: QAD output converted to XML:**

The screenshot shows a software interface with two main panes. The left pane, titled "amount of input is 3693 bytes and 112 lines", displays a text-based invoice. The right pane, titled "there are 5013 bytes in the output", displays the XML representation of the same invoice. The XML structure is shown as a tree view with expandable nodes.

**Text Output (Left Pane):**

```

18
19
20 Sales Order: 12409 Ship Date: 10/29/10
21 Order Date: 05/10/10 Purchase Order: 0060103
22 Salesperson(s): 10 Ship To: 401575
23 Credit Terms: NET 30 Ship Via: hertz truck line BOL: 000441
24
25 BAL DOE IN 30 DAYS FOB Point: FACTORY
26
27 Resale: 461495
28 Remarks: Fed Ex Ground Account # 42771956
29
30 CONSIGNEE:
31 Minisoft CSP
32 1024 First Street
33 Snohomish WA 98290
34 800-682-0200
35
36 Item Number UM Shipped Qty B/O Tax Price Extended Price
37 -----
38 04-135 EA 126.0 0.0 No 56.05 7,062.30
39 Customer Item: 413-25
40 CUSTOMER P 413-25
41
42 ***** CONTINUED *****
43
44
45
46
47
48
49
50
51
52
53 <ff>
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87

```

**XML Output (Right Pane):**

```

Invoice ( 223075 )
Revision ( 0 )
InvoiceDate ( 10/29/10 )
PrintDate ( 11/03/10 )
BillTo ( 201575 )
SoldTo ( 201575 )
BillToCompany ( Minisoft Inc. )
ShipToCompany ( Minisoft Inc. )
BillToAddressLn1 ( Snohomish WA 98290 )
ShipToAddressLn1 ( Snohomish WA 98290 )
Remarks
DetailLine
Totals

```

at row 0 column 0

Test Results Test Log Test Log 2

Verbose Test Save Load File... Save As... Exit

## **When should the preprocessor be used?**

When Print Output is formatted in a way that includes variably-sized blocks of data, the preprocessor can be used to enhance output that needs to be independent of the input page's conversion. Ordinarily, variable blocks would be Line items with multiple Lines of description.

## **When should the preprocessor not be used?**

Already XML - The Print Output needs to be in text or PCL format. The Text to XML conversion accepts text as input. Source data in XML, PDF, or image formats are not applicable to this process.

Too many pages - The Text to XML process keeps the working document in memory. Most systems will only handle a few thousand pages in a single Print Output without problems. If the Print Output is too large, use an output Bursting project to keep the size reasonable.

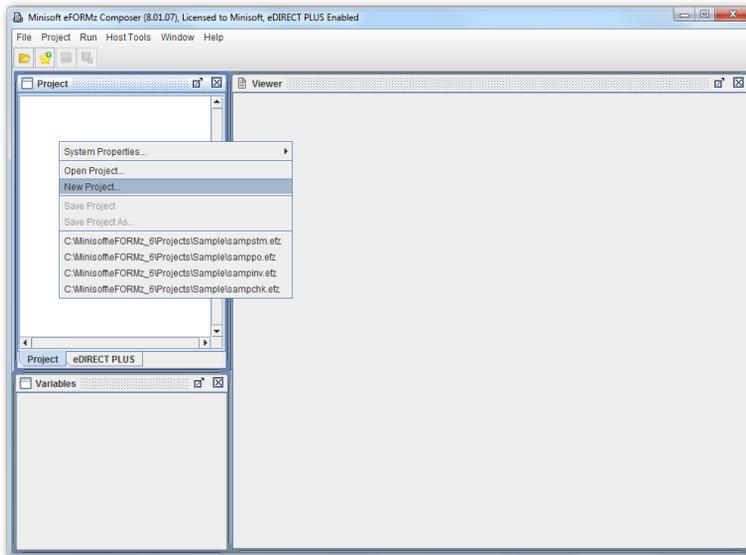


# *Configuring Text to XML*

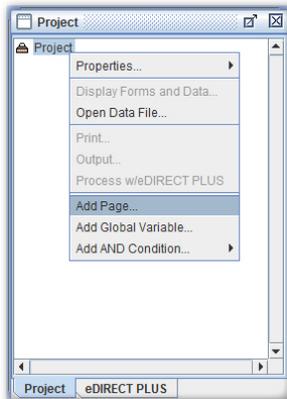
# Setting a Preprocessor for Text to XML Conversion

To implement the Text to XML Converter:

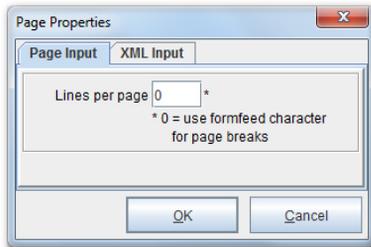
1. Create a new eFORMz Project by *right clicking* in the Project Window and selecting the *New Project* option.



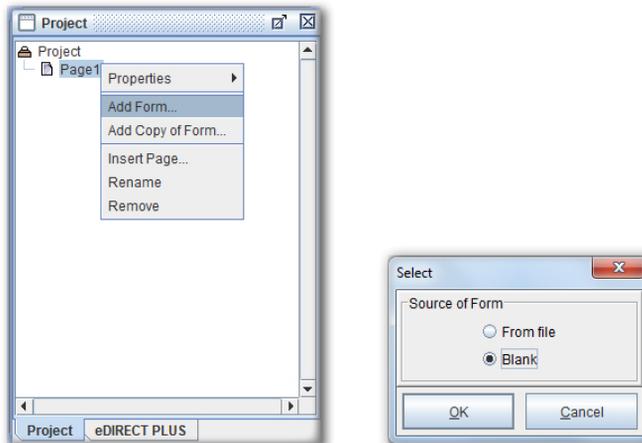
2. Next, *right click* the new Project name from the Project Viewer and choose *Add Page*.



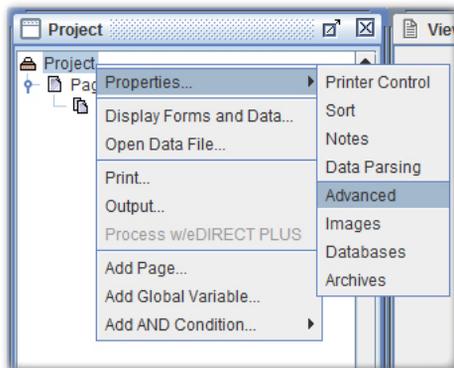
- The Page Properties dialog box will display. Click *OK* to proceed.



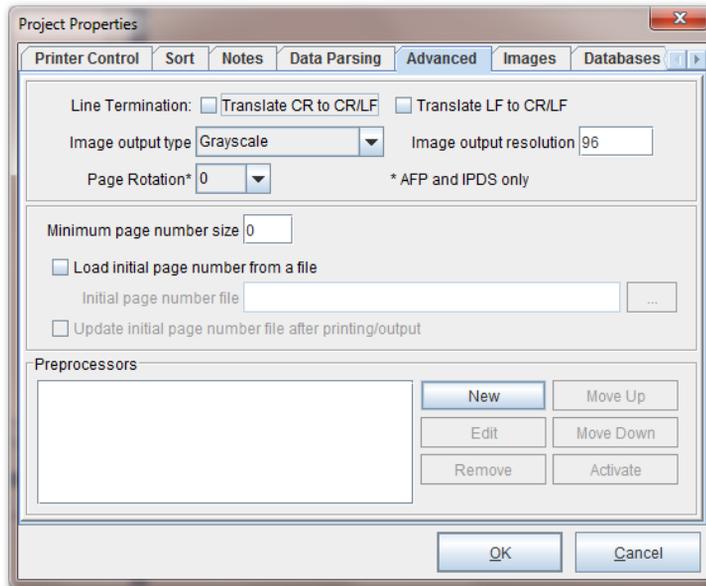
- To add a Form, *right click* the new Page and pick the *Add Form* option. Select *Blank*.



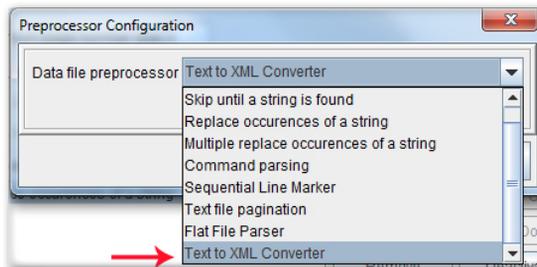
- Right click* the Project and select *Properties > Advanced*.



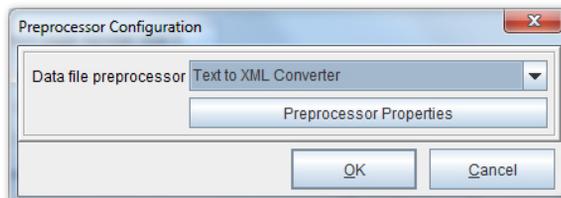
6. From the *Advanced tab > Preprocessors* section, select *New*.



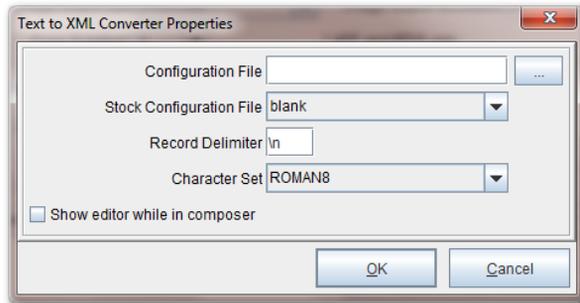
7. Using the *Data file preprocessor* drop-down menu, select the *Text to XML Converter* option.



8. Edit the Preprocessor Properties by selecting *Preprocessor Properties*.



9. The Text to XML Converter Properties dialog box displays.



Options Available:

**Configuration File:** The file to configure the initial settings of the Text to XML Editor. This is an optional parameter and can be left blank.

**Stock Configuration File:** A pre-existing file that configures the initial settings of the Text to XML Editor.

**NOTE:** There are three stock options: *RedPrairie Ship Confirmation*, *RedPrairie Order Confirmation* and *blank*.

The RedPrairie Ship Confirmation structures the initial settings of the Text to XML Editor to handle a shipping confirmation generated by the RedPrairie Direct Commerce system. The RedPrairie Order Confirmation structures the initial settings of the Text to XML Editor to handle an order confirmation generated by the RedPrairie Direct Commerce system. These files can be modified with the Text to XML Editor, but by specifying the basic conditions, they can save the user significant time. Note: Default is *blank*.

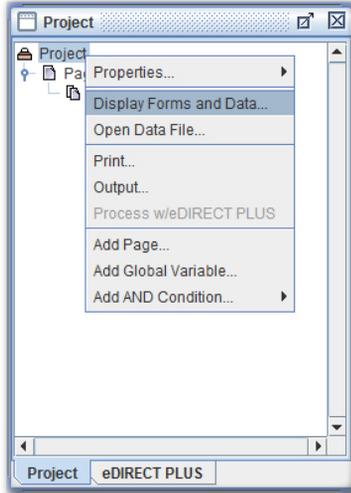
**Record delimiter:** Separates groups of fields by specifying the boundaries of the data stream. Default is a linefeed (\n = new line).

**Character Set:** All numbers, letters and symbols associated with a given coding system.

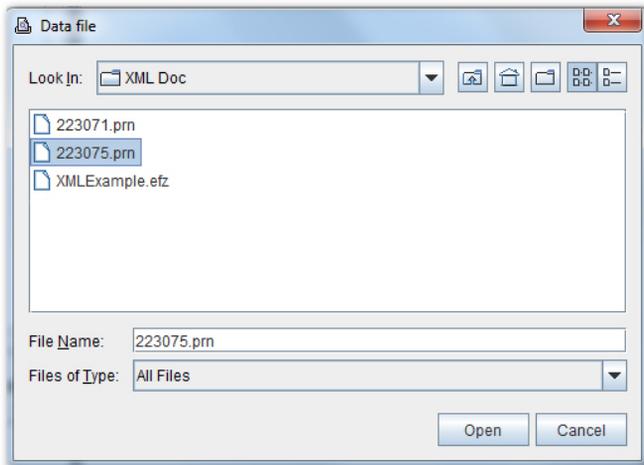
**Show editor while in composer:** When checked, displays the Text to XML Editor in the eFORMz Composer (Window > Text to XML Editor). Default is unchecked.

10. From the Stock Configuration File drop down, select *blank* and then select *OK*.

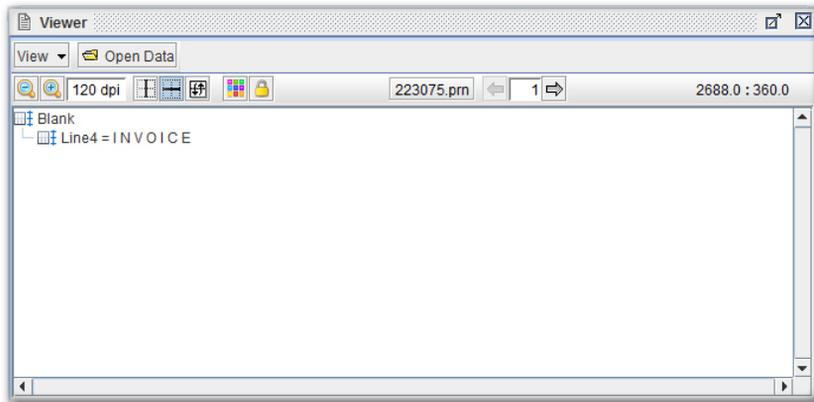
- Return to the eFORMz Composer. Right click the *Project* and select *Display Forms and Data*.



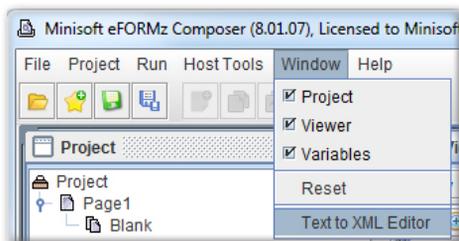
- Choose the file to convert to XML and click *OK*.



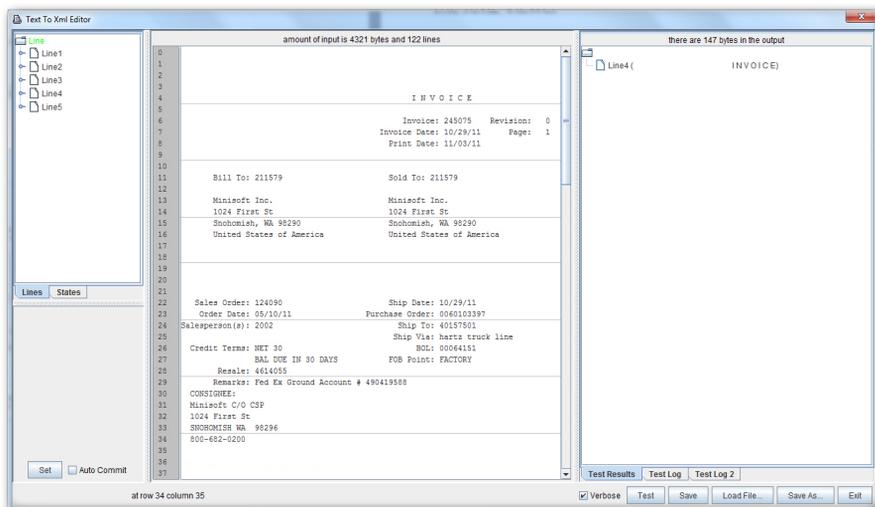
- The file will display as an XML file format in the Input Data Viewer.



- Once a given file has been converted, it can now be edited and configured for XML. To employ the Text to XML Editor, navigate to the *Window menu* and open the *Text to XML Editor*.



- The Text to XML Editor will appear in a new window.



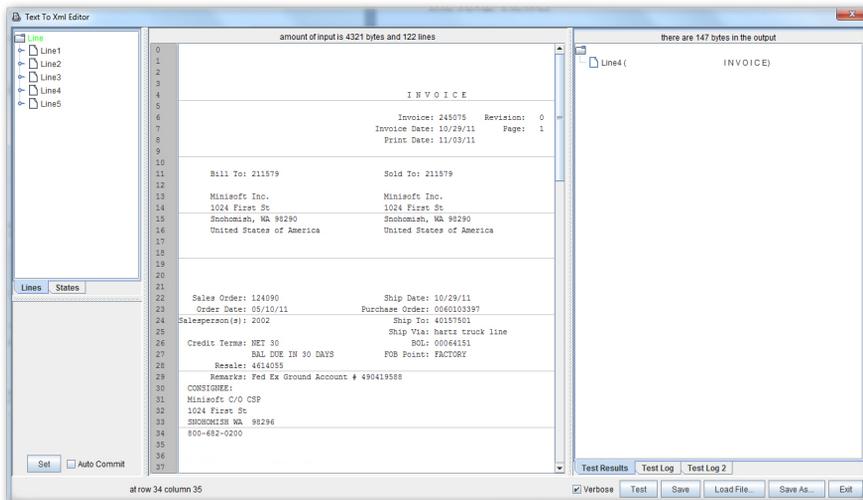
# The Text to XML Editor

Once a given file has been converted, it can now be edited and configured for XML. To employ the Text to XML Editor:

1. Navigate to the *Window menu* and open the *Text to XML Editor*.

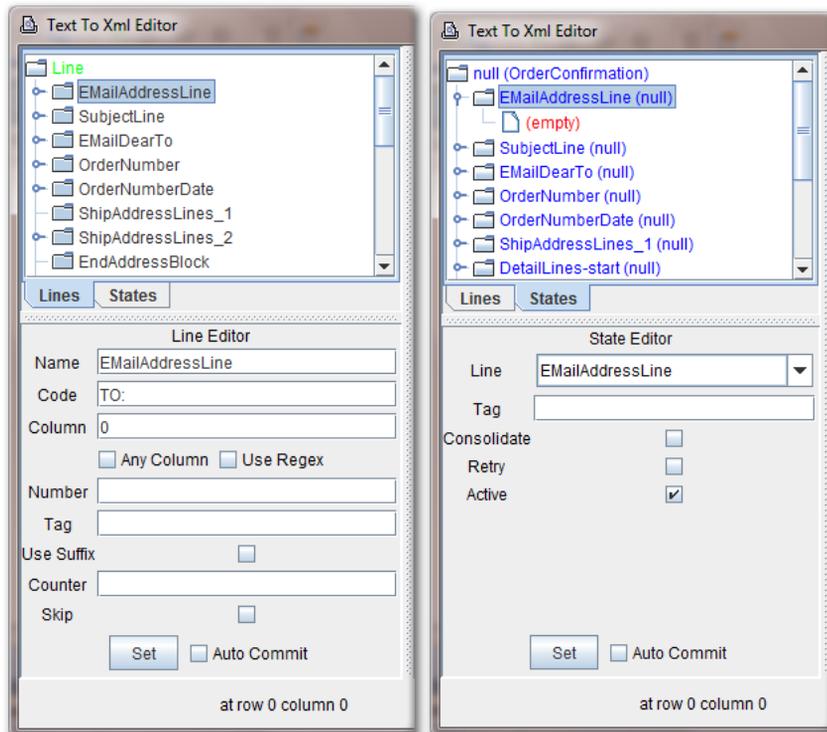


2. The Text to XML Editor will materialize in a new window.
3. The Text to XML Editor has three main window panes:
  - ◆ Line and State Viewer
  - ◆ Input Data Viewer
  - ◆ XML Data Viewer



## Line and State Viewer

The Lines and States Viewer allows you to edit and configure XML tags. Lines Configured in the Input Data Viewer will automatically display in the Lines Viewer. The Line at the top is the root element. A Line finds data in the input file. It can find a specific Line in the file by a number or by definition. The bottom, the Line Editor, shows the parameters of the particular Line.



## Adding Lines & Parameters in the Text to XML Editor

Having been introduced to the Text to XML Editor, the process of converting data to XML will necessitate the creation of Lines and Parameters.

1. To create a line, place the cursor at the Line that data will be found on in the Data Viewer.



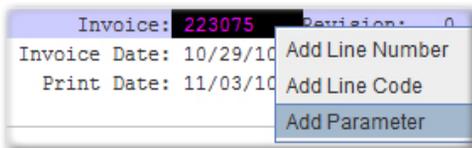
2. Right click and select Add Line Number.



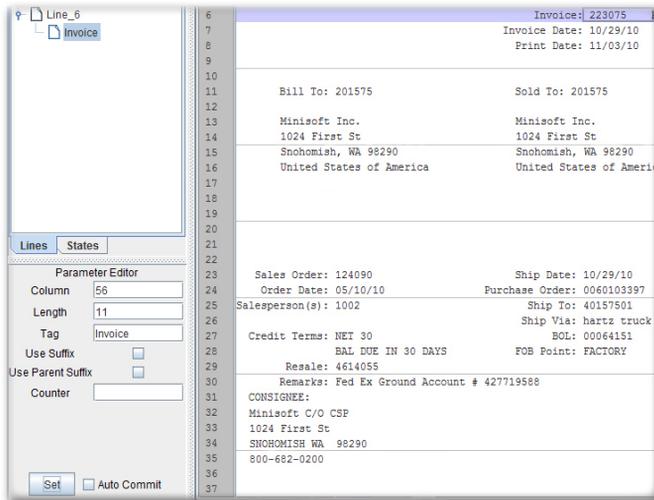
3. The Line Number will appear.



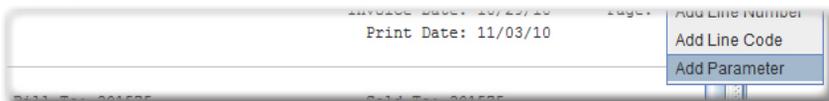
4. Parameters can now be added to the Line. Highlight the Invoice number all the way to the word Revision. Note: Highlighting the Parameter until the instance of another Parameter or label is a general practice that will ensure that all incoming data is picked up.



- The Parameter will appear with a tag of NewParameter. In the Tag field of the Parameter Editor, enter the name of the Parameter you want displayed in the XML output. Set the changes. Note: To further aid the process of picking up all of the incoming data for a given parameter, the Length field can be adjusted. If the Parameter's potential length isn't fully highlighted initially, specifying the Length acts as a good safeguard for finding all of the incoming data associated with that Parameter.



- To add another Parameter on the same Line, first select the Line icon. Then, highlight the Revision number, right click and pick the Add Parameter option.



- Specify the Tag and verify the Length in the Parameter Editor. Set the Changes. The Revision Parameter will display.

**Parameter Editor**

Column:

Length:

Tag:

Use Suffix:

Use Parent Suffix:

Counter:

Auto Commit

Line_6	3	
Invoice	4	I N V O I C E
Revision	5	
	6	Invoice: 223075    Revision: <input type="text" value="0"/>

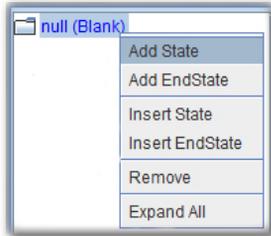
- Continue adding Lines and Parameters.

Line	11	amount of input is 4321 bytes and 122 lines	
Line_6	12	Bill To: 2015/5	Sold To: 2015/5
Line_7	13	Minisoft Inc.	Minisoft Inc.
Line_8	14	1024 First St	1024 First St
Line_11	15	Snohomish, WA 98290	Snohomish, WA 98290
Line_13	16	United States of America	United States of America
Line_14	17		
Line_15	18		
Line_16	19		
Line_17	20		
Line_23	21		
Line_24	22	Sales Order: 124090	Ship Date: 10/29/10
Line_25	23	Order Date: 05/10/10	Purchase Order: 0060103397
Line_26	24	Salesperson(s): 1002	Ship To: 40157501
Line_27	25		Ship Via: hartz truck line
Line_28	26	Credit Terms: NET 30	BOL: 00064151
Line_29	27	BAL DUE IN 30 DAYS	FOB Point: FACTORY
	28		
	29	Resale: 4614055	

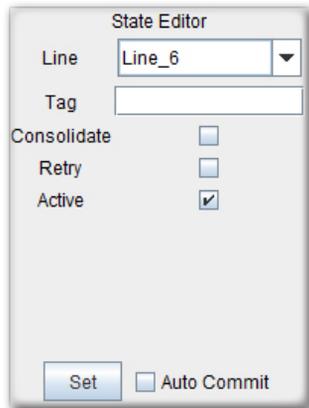
## Adding States in the Text to XML Editor

While Lines find the data, and Parameters describe it, States put the Lines and Parameters to work. To finish the XML conversion process, the Lines and Parameters that have been created must be referenced by States.

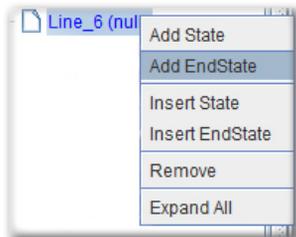
1. Switch to the States tab. Right click null (Blank) and select Add State.



2. Select Line 6 from the Line drop-down menu. Click Set.

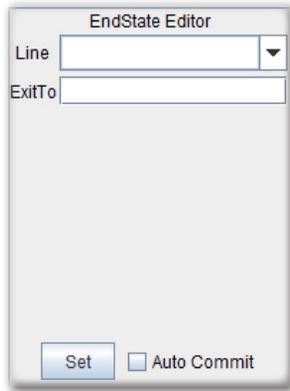


3. Right click the State and select Add EndState.

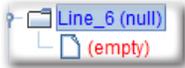


## SECTION 1

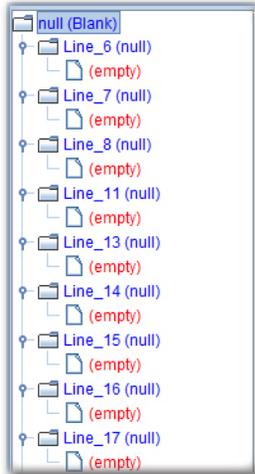
- Click Set. Note: An empty setting means that an explicit line is only going to be processed once.



- The State with the information from Line\_6 will display.



- Continue adding States and EndStates.

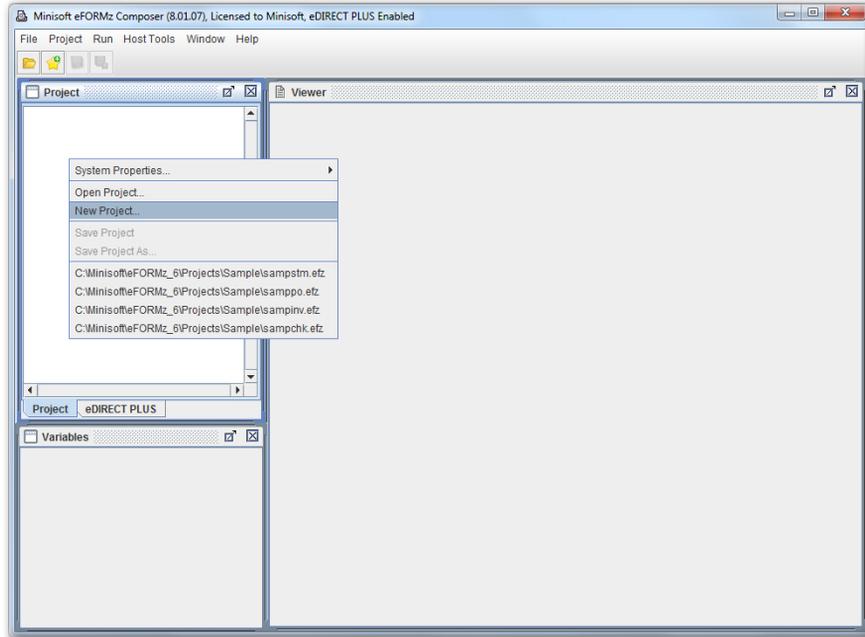




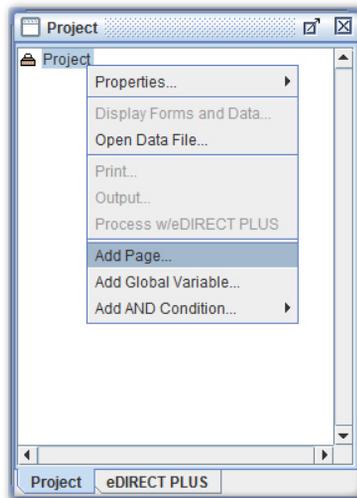
# *Configuring for QAD*

Create a new eFORMz Project:

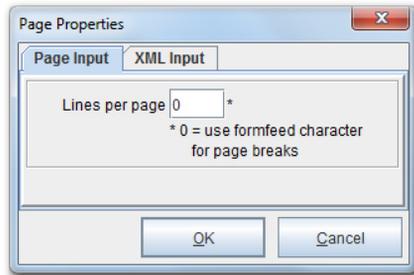
1. *Right clicking* in the Project Window and selecting the *New Project* option.



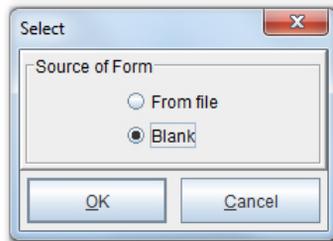
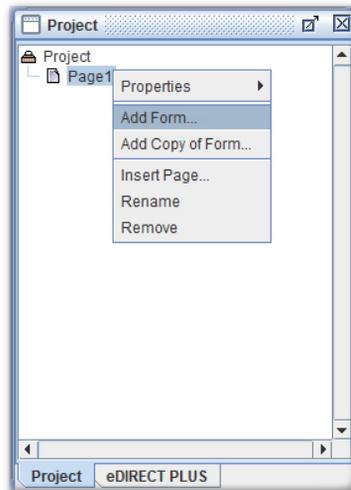
2. Next, *right click* the new Project and choose *Add Page*.



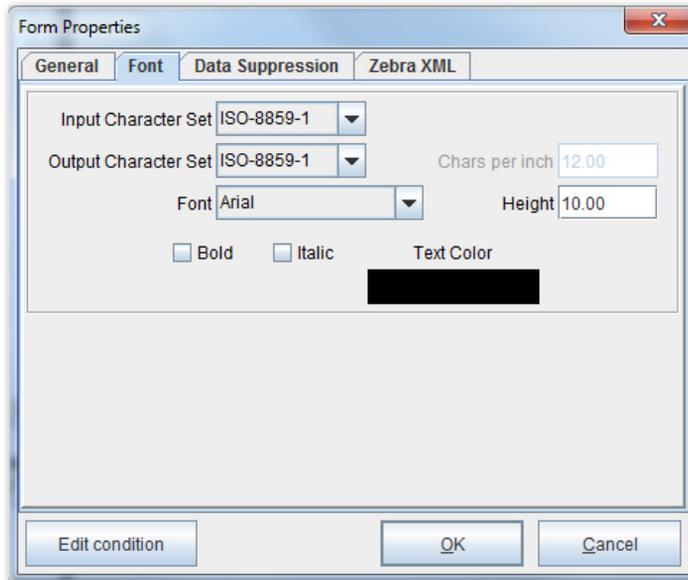
3. The *Page Properties* dialog box will display. Click *OK* to proceed.



4. To add a Form, right click the new *Page* and pick the *Add Form* option. Select *Blank*.

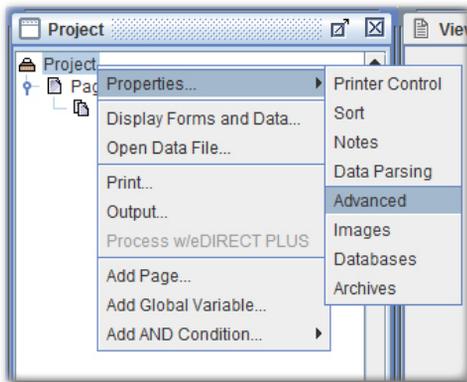


- The Form Properties dialog box will appear. Navigate to the *Fonts* tab. Set the *Input Character Set* and the *Output Character Set* to *ISO-8859-1*. In the Font field, select *Arial* from the drop-down menu and set the *Height* to *10*.

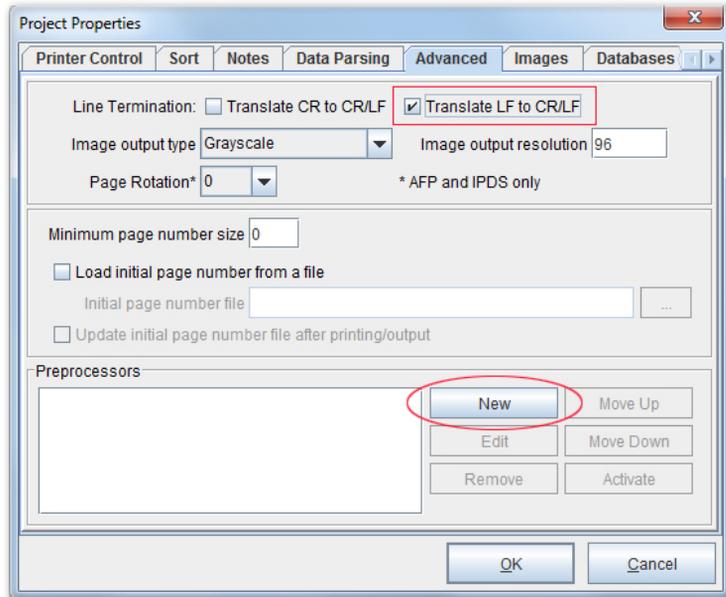


**TIP:** By using ISO-8859-1, there is continuity between input and output regardless of platform.

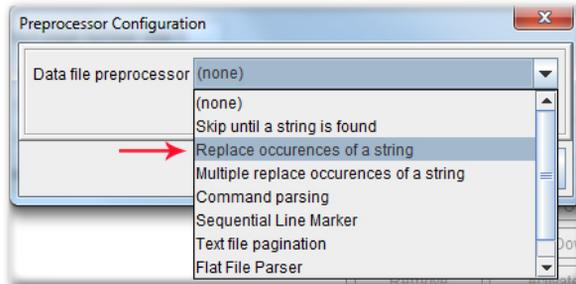
- Right click the Project and select *Properties > Advanced*.



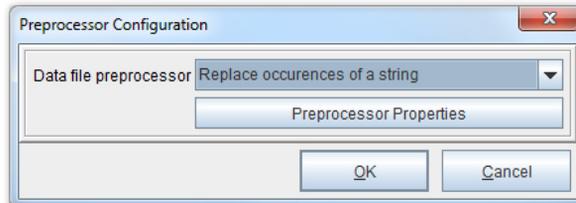
7. The Project Properties dialog box will display. First, check the *Translate LF to CR/LF* option in the Line Termination field. Then, in the Preprocessors section, select *New*.



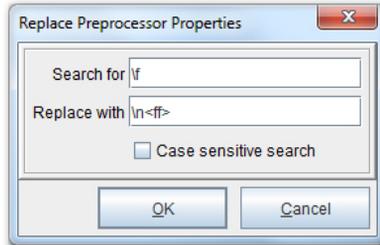
8. The Preprocessor Configuration dialog box will appear. Utilizing the Data file preprocessor drop-down menu, select *Replace occurrences of a string*.



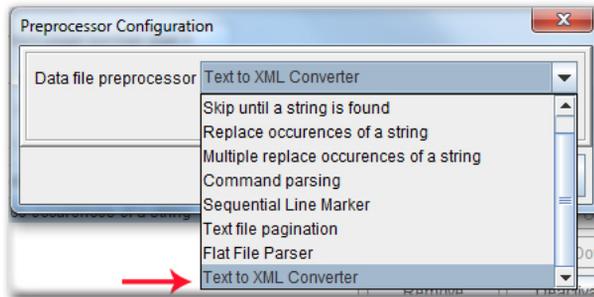
9. With the choice made, edit the *Preprocessor Properties*.



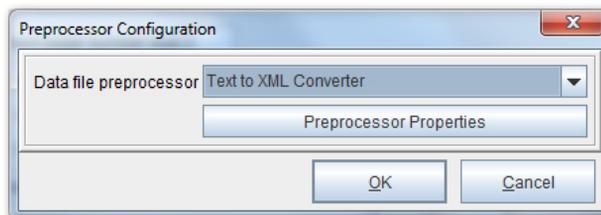
- The Replace Preprocessor Properties dialog box comes into view. In the *Search for* field, type `\f` and in the *Replace with* field, type `\n<ff>`. This will replace all occurrences of FormFeed with a new Line and the characters `<ff>`. Select *OK*.



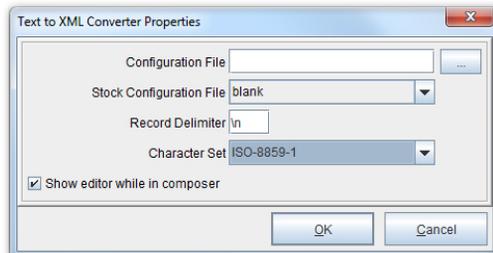
- Using the *Data file preprocessor* drop-down menu once more, select the *Text to XML Converter* option.



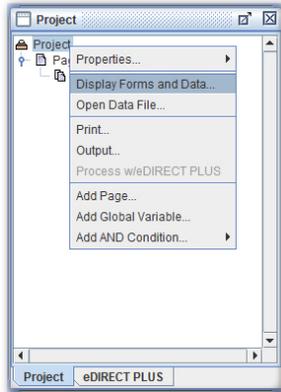
- Edit the *Preprocessor Properties*.



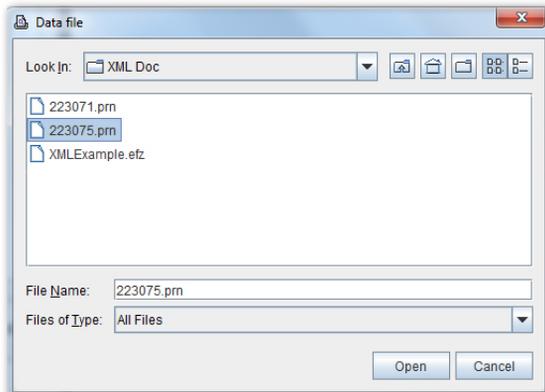
- The Text to XML Converter Properties dialog box comes into view. From the *Character Set* drop-down menu, select the *ISO-8859-1* option. Check the *Show editor while in composer* box and then click *OK*.



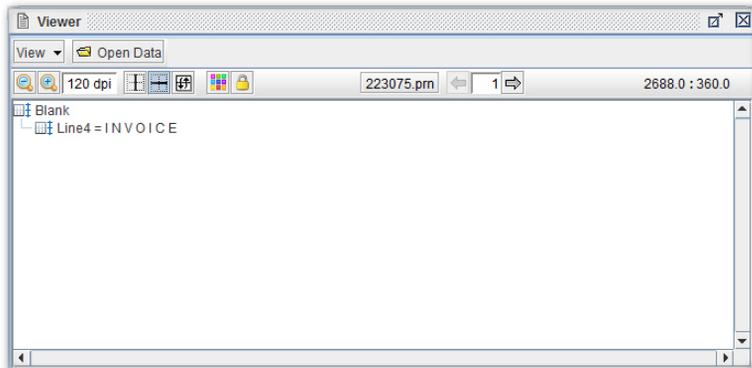
14. Return to the eFORMz Composer. Right click the *Project* and select *Display Forms and Data*.



15. Choose the file to convert to XML and click OK.



16. The file will display as an XML file in the *Input Data Viewer*.



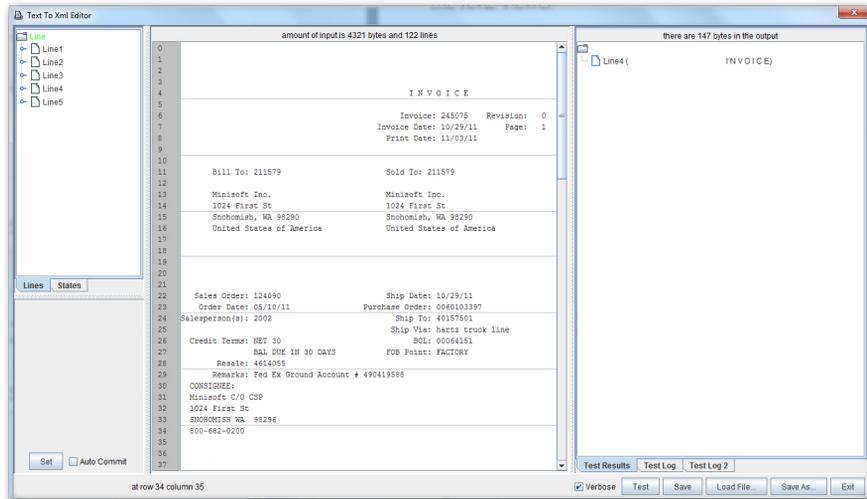
## SECTION 2

Once a given file has been converted, it can now be edited and configured for XML. To employ the Text to XML Editor:

1. Navigate to the *Window* menu and open the *Text to XML Editor*.



2. The Text to XML Editor will materialize in a new window.



# Text to XML Editor

## Adding Lines & Parameters

The process of converting data to XML will first require the creation of Lines and Parameters.

1. To create a line, place the cursor at the Line that data will be found on in the Data Viewer.



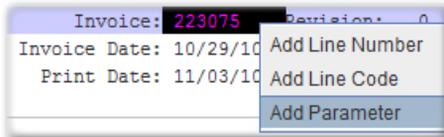
2. *Right click* and select *Add Line Number*.



3. The Line Number will appear in the Line and Parameter Viewer.



4. Parameters can now be added to the Line. Highlight the Invoice number all the way to the word Revision. Note: Highlighting the Parameter until the instance of another Parameter or label is a general practice that will ensure that all incoming data is picked up.



## SECTION 2

- The Parameter will appear with a tag of NewParameter. In the Tag field of the Parameter Editor, enter the name of the Parameter you want displayed in the XML output. Set the changes by selecting *Set*.

The screenshot shows a software interface with a 'Parameter Editor' window open for 'Line\_6'. The editor has the following fields:

- Column: 56
- Length: 11
- Tag: Invoice
- Use Suffix:
- Use Parent Suffix:
- Counter: [empty]

At the bottom of the editor are buttons for 'Set' and 'Auto Commit'. The background shows an invoice summary for Invoice 223075, including dates, bill-to/sold-to information, and shipping details.

**NOTE:** To further aid the process of picking up all of the incoming data for a given parameter, the Length field can be adjusted. If the Parameter's potential length isn't fully highlighted initially, specifying the Length acts as a good safeguard for finding all of the incoming data associated with that Parameter.

- To add another Parameter on the same Line, first select the Line icon. Then, highlight the Revision number, right click and pick the Add Parameter option.

The screenshot shows the software interface with the 'Parameter Editor' window open. The 'Revision' field is highlighted, and a context menu is open with the 'Add Parameter' option selected.

- Specify the Tag and verify the Length in the Parameter Editor. Set the Changes. The Revision Parameter will display.

**Parameter Editor**

Column

Length

Tag

Use Suffix

Use Parent Suffix

Counter

Auto Commit

Line_6	3		
Invoice	4		
Revision	5		
	6		

I N V O I C E	
Invoice: 223075	Revision: 0

- Continue adding Lines and Parameters:

Line		amount of input is 4321 bytes and 122 lines	
Line_6	11	Bill To: 201575	Sold To: 201575
Line_7	12		
Line_8	13	Minisoft Inc.	Minisoft Inc.
Line_11	14	1024 First St	1024 First St
Line_13	15	Snohomish, WA 98290	Snohomish, WA 98290
Line_14	16	United States of America	United States of America
Line_15	17		
Line_16	18		
Line_17	19		
Line_23	20		
Line_24	21	Sales Order: 124090	Ship Date: 10/29/10
Line_25	22	Order Date: 05/10/10	Purchase Order: 0060103397
Line_26	23	Salesperson(s): 1002	Ship To: 40157501
Line_27	24		Ship Via: hartz truck line
Line_28	25	Credit Terms: NET 30	BOL: 00064151
Line_29	26	BAL DUE IN 30 DAYS	FOB Point: FACTORY
	27	Resale: 4614055	

# Troubleshooting with Lines & Parameters

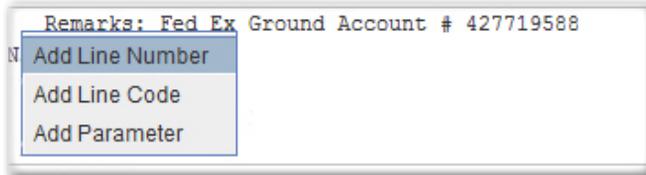
When using Lines and Parameters to find and describe data, difficulties may arise. Some common problems users face are how to deal with comments, detail lines and blank spaces. An approach to these portions of a data file is the subject of this section.

## Comments & Remarks

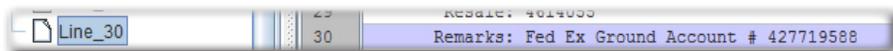
1. First, navigate to the Line where comments begin. Establish the cursor as shown before.



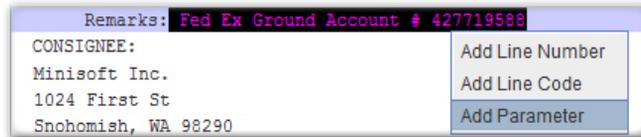
2. *Right click* the Line and select *Add Line Number*.



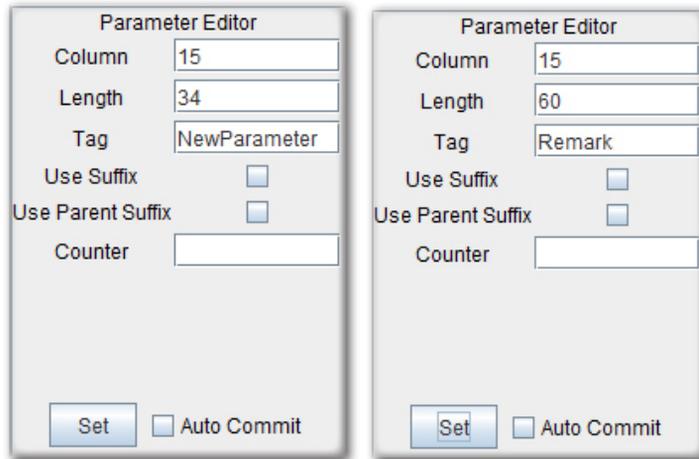
3. The Line will display.



4. Highlight the data after Remarks: and select *Add Parameter*.



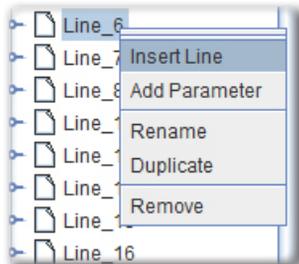
- In the Parameter Editor, enter a name in the Tag field (i.e. Remark) and increase the Length of the Parameter.



**NOTE:** The box outlining the Parameter will not change in appearance, but it will conform to the Length specified in the Parameter Editor.

Remarks: Fed Ex Ground Account # 427719588

- Highlight any Line, right click and select *Insert Line*.





9. Right click the Line and select *Add Parameter*.



10. In the Parameter Editor, set the Column to 2, the Length to 78 and the Tag to Remark.

Parameter Editor

Column

Length

Tag

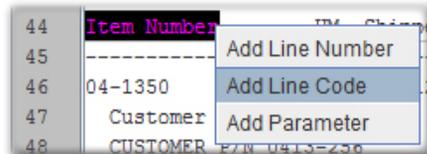
Use Suffix

Use Parent Suffix

Counter

Auto Commit

11. Next, scroll to the Line which begins the detail lines. Highlight the first instance of data on the Line (i.e.: Item Number), *right click* and select *Add Line Code*.



**NOTE:** Add Line Code will not reference the particular Line Number, but any line of the data that matches the value entered in Line Code. Multiple lines can match the value.

## SECTION 2

- In the Line Editor, enter the Name (Ex. CommentsEnd) and set the change. This Line is intended to end the Comments section. Note: As no Parameter will be defined, the data will not be included as XML.

**Line Editor**

Name

Code

Column

Any Column  Use Regex

Number

Tag

Use Suffix

Counter

Skip

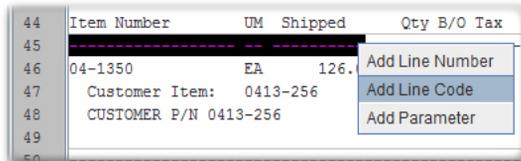
Auto Commit

**NOTE:** In the Code field, only Item Number appears, as it was the lone piece of data highlighted. However, the instance of Item Number on a particular Line will make all other data following on that Line subject to the same actions taken by States.

44	Item Number	UM	Shipped	Qty B/O Tax	Price	Exter
84	Item Number	UM	Shipped	Qty B/O Tax	Price	Extended Pric

## Detail Lines

- To begin adding detail lines, scroll to the line below the CommentsEnd Line. Highlight the dashes from the first column through Shipped, or enough so that there are spaces between the groups of dashes, right click and choose Add Line Code.



- In the Line Editor, enter the Name (Ex. DetailsStart) and set the change. This Line will mark the beginning of the detail lines. Note: As there is no Parameter, the dashes will not be included as XML. This Line acts as a marker and its contents will be acted upon by States in the same fashion as the CommentsEnd Line.

**Line Editor**

Name:

Code:

Column:

Any Column  Use Regex

Number:

Tag:

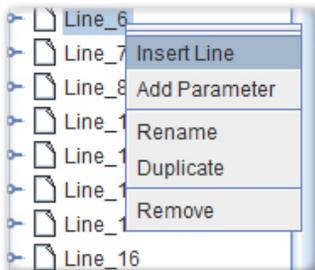
Use Suffix:

Counter:

Skip:

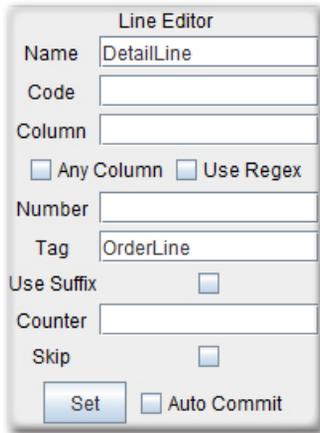
Auto Commit

- Right click any Line and select Insert Line.



## SECTION 2

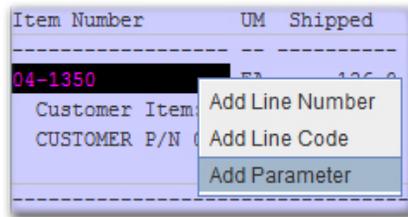
4. Enter a *Name* (i.e. DetailLine) and set the *Tag* field (i.e. OrderLine).



The Line Editor dialog box contains the following fields and options:

- Name: DetailLine
- Code: [Empty]
- Column: [Empty]
- Any Column  Use Regex
- Number: [Empty]
- Tag: OrderLine
- Use Suffix:
- Counter: [Empty]
- Skip:
- Buttons: Set,  Auto Commit

5. With the DetailLine defined, highlight the first instance of data and select *Add Parameter*.



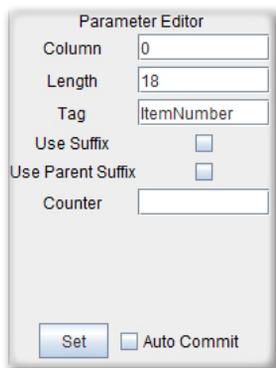
Item Number	UM	Shipped
04-1350		
Customer Item		
CUSTOMER P/N		

A context menu is open over the first data row, showing the following options:

- Add Line Number
- Add Line Code
- Add Parameter

**TIP:** Highlight to meet the dashes, as they indicate the maximum amount of data that will occur for a given field. As a result, Parameters will pick up all of the incoming data.

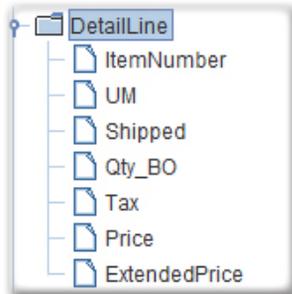
6. In the Parameter Editor, set the *Tag* to correspond with the data being described (i.e. ItemNumber).



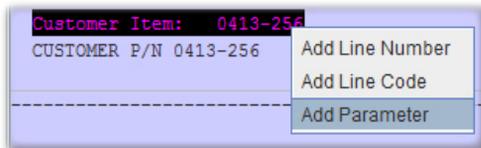
The Parameter Editor dialog box contains the following fields and options:

- Column: 0
- Length: 18
- Tag: ItemNumber
- Use Suffix:
- Use Parent Suffix:
- Counter: [Empty]
- Buttons: Set,  Auto Commit

7. Continue adding Parameters in this manner until reaching any descriptions.



8. With a description, highlight it, *right click* and select *Add Parameter*. Note: There is no need to highlight from Column o.



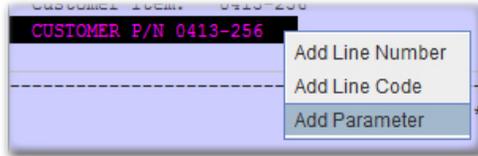
9. In the Parameter Editor, make the Length 60 and enter a value in the Tag field (i.e. Description) and set the changes.

Parameter Editor	
Column	<input type="text" value="2"/>
Length	<input type="text" value="60"/>
Tag	<input type="text" value="Description"/>
Use Suffix	<input type="checkbox"/>
Use Parent Suffix	<input type="checkbox"/>
Counter	<input type="text"/>
<input type="button" value="Set"/> <input type="checkbox"/> Auto Commit	

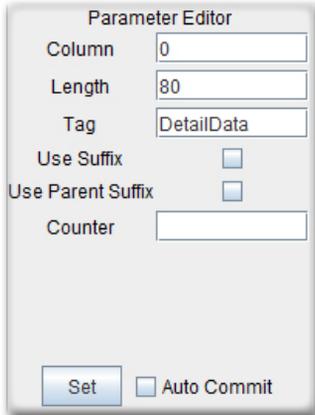
**NOTE:** 60 should suffice for picking up all of the description. As highlighting beyond the present data may not be possible, always adjust the Length to the appropriate level in the Parameter Editor.

## SECTION 2

10. Highlight the final Line, *right click* and pick the *Add Parameter* option.



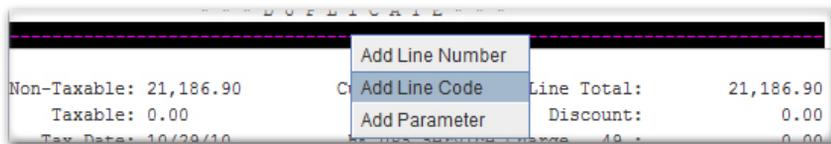
11. In the Parameter Editor, make the Length 80 and enter a value in the Tag field (i.e. DetailData) and set the changes.

A screenshot of the 'Parameter Editor' dialog box. It contains several input fields and checkboxes. The 'Column' field is set to '0', 'Length' is '80', and 'Tag' is 'DetailData'. There are checkboxes for 'Use Suffix' and 'Use Parent Suffix', both of which are unchecked. A 'Counter' field is empty. At the bottom, there is a 'Set' button and an 'Auto Commit' checkbox, which is also unchecked.

**NOTE:** 80 should suffice for picking up any additional, miscellaneous detail items. As highlighting beyond the present data may not be possible, always adjust the Length to the appropriate level in the Parameter Editor.

12. Add a Line Code for the Line of dashes below the DetailData Line by highlighting all of the dashes, *right clicking* and selecting *Add Line Code*.

Note: These dashes will not be included as XML, as no Parameter will describe the Line.

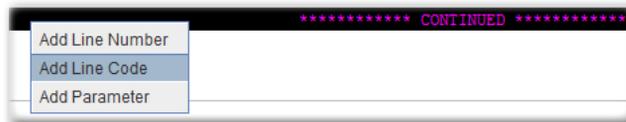


13. Enter a *Name* (i.e. Dashes) and set the changes.

The 'Line Editor' dialog box is shown with the following fields and options:

- Name: Dashes
- Code: -----
- Column: 0
- Any Column  Use Regex
- Number: [empty]
- Tag: [empty]
- Use Suffix:
- Counter: [empty]
- Skip:
- Buttons: Set,  Auto Commit

14. Add a *Line Code* for the Line below the Dashes Line by highlighting the whole line, right clicking and selecting *Add Line Code*.



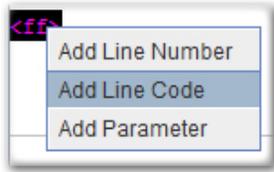
15. Enter a *Name* (i.e. Continued) and set the changes.

The 'Line Editor' dialog box is shown with the following fields and options:

- Name: Continued
- Code: CONTINUED \*\*\*\*\*
- Column: 0
- Any Column  Use Regex
- Number: [empty]
- Tag: [empty]
- Use Suffix:
- Counter: [empty]
- Skip:
- Buttons: Set,  Auto Commit

## SECTION 2

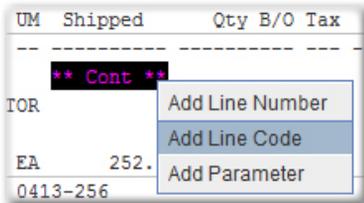
16. Add a Line Code for the Line with the <ff> tag by highlighting the whole line, right clicking and selecting *Add Line Code*.



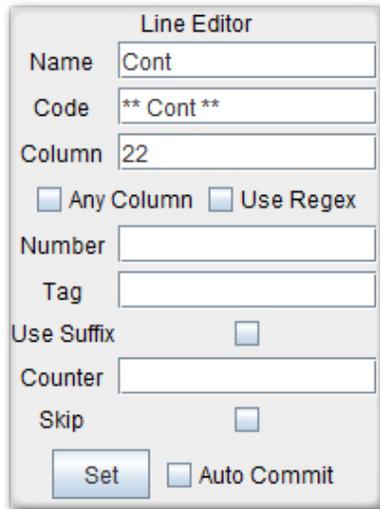
17. Enter a Name (i.e. FormFeed) and set the changes.

A screenshot of the 'Line Editor' dialog box. The 'Name' field contains 'FormFeed', the 'Code' field contains '<ff>', and the 'Column' field contains '0'. There are checkboxes for 'Any Column' and 'Use Regex', both of which are unchecked. There are also fields for 'Number', 'Tag', 'Counter', and 'Skip', all of which are empty. A 'Set' button and an 'Auto Commit' checkbox are at the bottom.

18. Add a Line Code for the Line with the **\*\* Cont \*\*** data by highlighting it, right clicking and selecting *Add Line Code*.



19. Enter a *Name* (i.e. Cont) and set the changes.



The 'Line Editor' dialog box contains the following fields and options:

- Name: Cont
- Code: \*\* Cont \*\*
- Column: 22
- Any Column  Use Regex
- Number: [Empty]
- Tag: [Empty]
- Use Suffix:
- Counter: [Empty]
- Skip:
- Buttons: Set,  Auto Commit

20. Add a *Line Code* for the Line with the \*\*\* D U P L I C A T E \*\*\* data by highlighting it, right clicking and selecting *Add Line Code*.



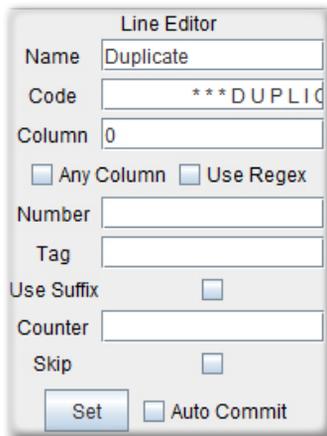
The screenshot shows a line item with the following details:

- Line Code: \*\*\* D U P L I C A T E \*\*\*
- Non-Taxable: 21,186.90
- Taxable: 0.00
- Currency: USD
- Line Total: [Empty]
- Discount: 0.00%

A context menu is open over the line code, with the following options:

- Add Line Number
- Add Line Code (highlighted)
- Add Parameter

21. Enter a *Name* (i.e. Duplicate) and set the changes.

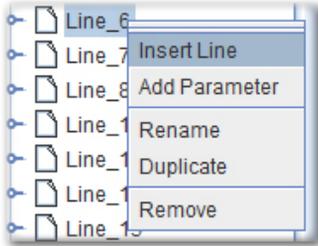


The 'Line Editor' dialog box contains the following fields and options:

- Name: Duplicate
- Code: \*\*\* D U P L I C
- Column: 0
- Any Column  Use Regex
- Number: [Empty]
- Tag: [Empty]
- Use Suffix:
- Counter: [Empty]
- Skip:
- Buttons: Set,  Auto Commit

## White Space

1. Highlight any Line, *right click* and select *Insert Line*.



2. In the Line Editor, begin by setting the Name field (i.e. Blank). In the Code field, enter the value of `^\s*$` and check the *Use Regex* box. Set *Column* to 0.

Line Editor

Name

Code

Column

Any Column  Use Regex

Number

Tag

Use Suffix

Counter

Skip

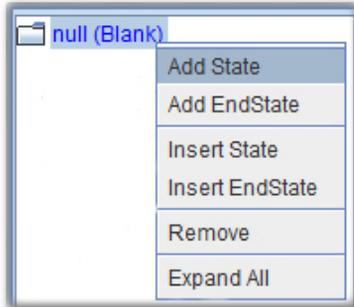
Auto Commit

**NOTE:** The regular expression in the Code field finds the Lines that contain nothing but white space.

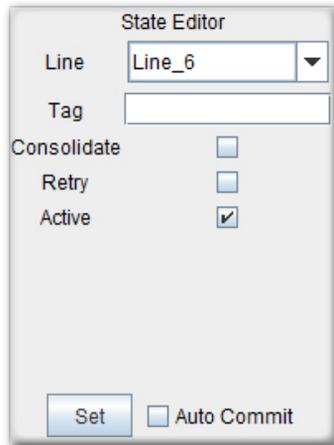
## Adding States in the Text to XML Editor

While Lines find the data, and Parameters describe it, States put the Lines and Parameters to work. In order to make the conversion to XML possible, the user must reference the Lines and Parameters they've created with States.

1. Switch to the *States* tab. Right click null (Blank) and select *Add State*.

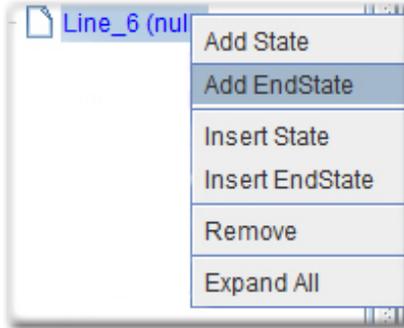


2. Select Line 6 from the *Line* drop-down menu. Click *Set*.

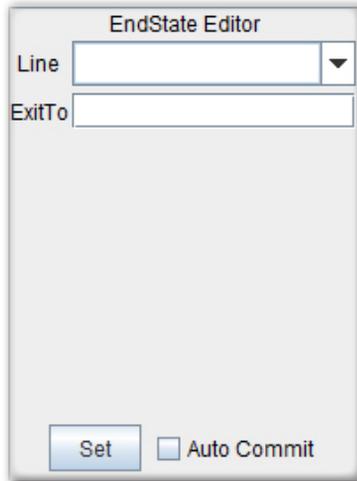


## SECTION 2

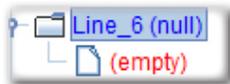
3. Right click the State and select *Add EndState*.



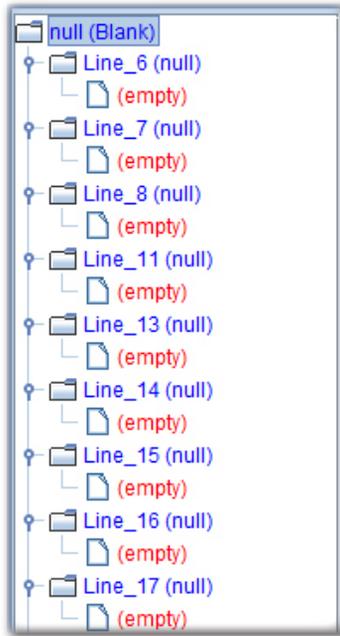
4. Click *Set*. Note: An empty setting means that an explicit line is only going to be processed once.



5. The State with the information from Line\_6 will display.



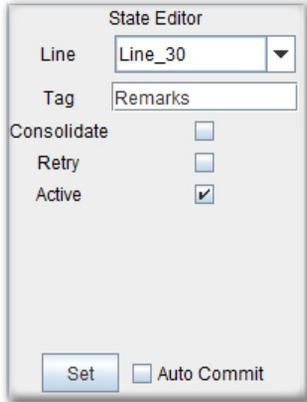
6. Continue adding States and EndStates.



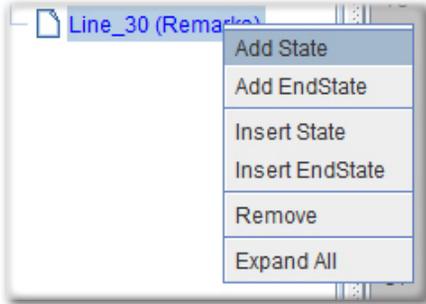
## Troubleshooting with States

Issues arise with States when nesting is involved, which is typically necessary with Line Code elements.

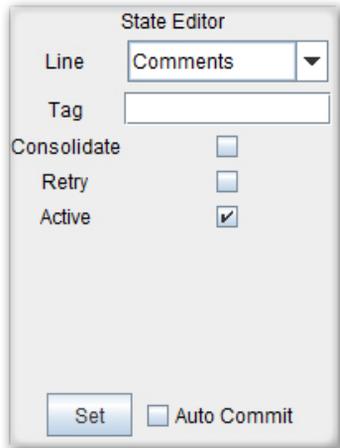
1. *Right click* any *State* and select *Insert State*. In the State Editor, choose Line\_30 from the drop-down menu and set the *Tag* to Remarks.



2. *Right click* the Line\_30 State and select *Add State*. Note: This will nest a state within an existing State. In other words, the existing State is a parent element and the nested State is a child element.

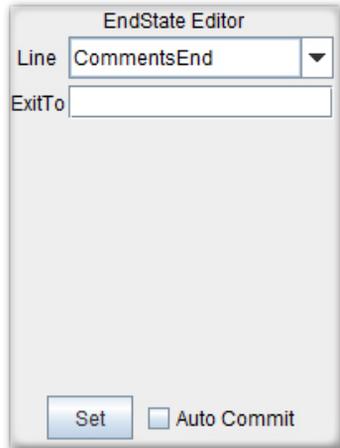


3. Choose Comments from the drop-down menu and click Set.



The screenshot shows the "State Editor" dialog box. It has a title bar "State Editor". Below the title bar, there is a "Line" field with a dropdown menu currently showing "Comments". Below that is a "Tag" field which is empty. There are three checkboxes: "Consolidate" (unchecked), "Retry" (unchecked), and "Active" (checked). At the bottom, there is a "Set" button and an "Auto Commit" checkbox (unchecked).

4. Right click Comments and select Add EndState. Select CommentsEnd.



The screenshot shows the "EndState Editor" dialog box. It has a title bar "EndState Editor". Below the title bar, there is a "Line" field with a dropdown menu currently showing "CommentsEnd". Below that is an "ExitTo" field which is empty. At the bottom, there is a "Set" button and an "Auto Commit" checkbox (unchecked).

5. Right click Line\_30 and select *Add EndState*. Select CommentsEnd.



EndState Editor

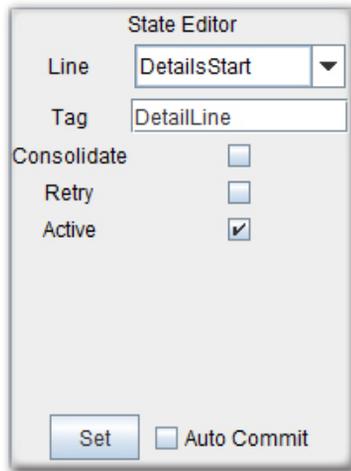
Line: CommentsEnd

ExitTo:

Set  Auto Commit



6. Right click any State and select *Insert State*. In the State Editor, choose DetailsStart from the drop-down menu and set the Tag to DetailLine.



State Editor

Line: DetailsStart

Tag: DetailLine

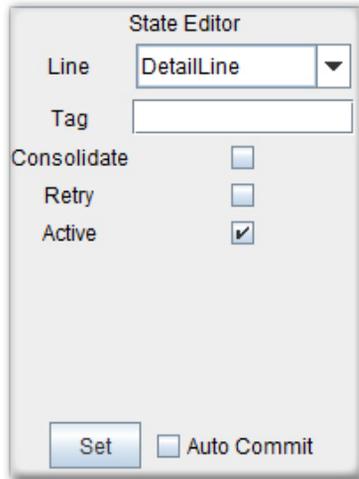
Consolidate

Retry

Active

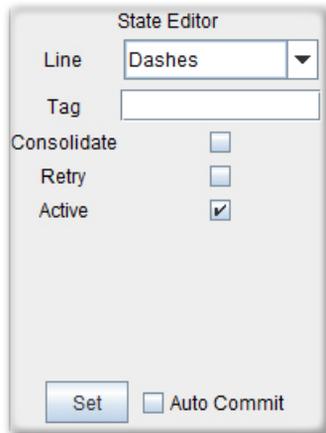
Set  Auto Commit

7. *Right click* the DetailsStart State and select *Add State*. In the State Editor, choose DetailLine from the drop-down menu and click *Set*.



The screenshot shows the 'State Editor' dialog box. The 'Line' dropdown menu is set to 'DetailLine'. The 'Tag' field is empty. The 'Consolidate' checkbox is unchecked, 'Retry' is unchecked, and 'Active' is checked. At the bottom, there is a 'Set' button and an unchecked 'Auto Commit' checkbox.

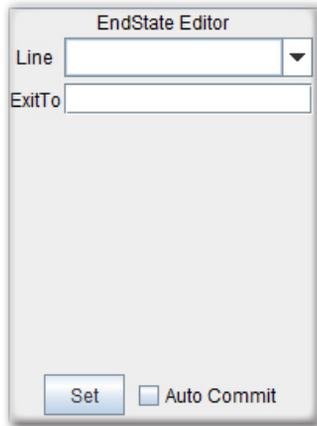
8. *Right click* the DetailsLine State and select *Add State*. In the State Editor, choose Dashes from the drop-down menu and click *Set*.



The screenshot shows the 'State Editor' dialog box. The 'Line' dropdown menu is set to 'Dashes'. The 'Tag' field is empty. The 'Consolidate' checkbox is unchecked, 'Retry' is unchecked, and 'Active' is checked. At the bottom, there is a 'Set' button and an unchecked 'Auto Commit' checkbox.

## SECTION 2

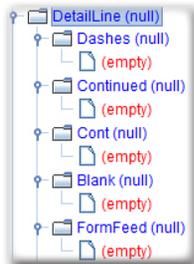
9. *Right click* the Dashes State and select *Add EndState*. Set the EndState Editor as empty.



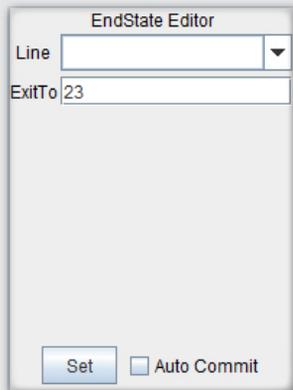
The EndState Editor dialog box is shown with the following fields and controls:

- Line:** A dropdown menu.
- ExitTo:** A text input field that is currently empty.
- Buttons:** A "Set" button and an "Auto Commit" checkbox (which is unchecked).

10. Continue adding States to the DetailLine State.



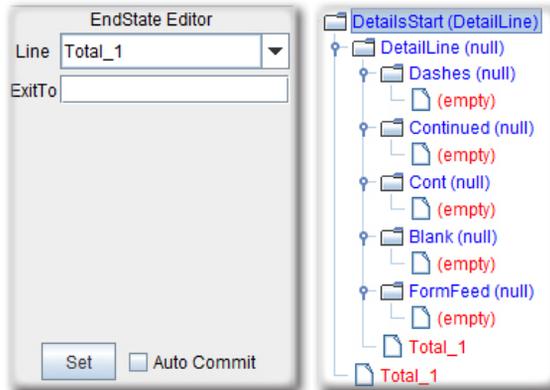
**NOTE:** To the FormFeed EndState Editor, enter the value of 23 to the ExitTo field. This will cause the parser to skip the next 23 lines.



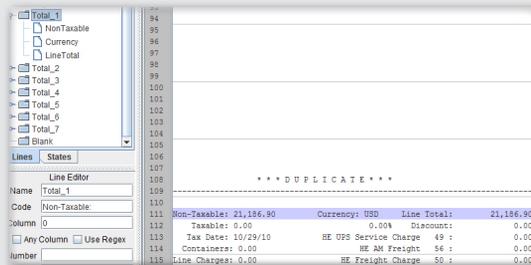
The EndState Editor dialog box is shown with the following fields and controls:

- Line:** A dropdown menu.
- ExitTo:** A text input field containing the value "23".
- Buttons:** A "Set" button and an "Auto Commit" checkbox (which is unchecked).

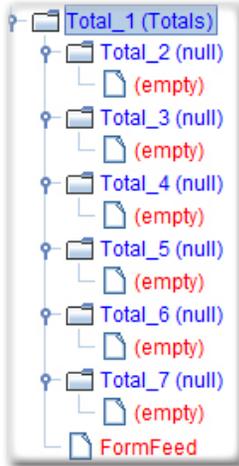
11. *Right click* the DetailsLine State and select *Add EndState*. Pick Total\_1 from the drop-down menu. Add the same EndState for the DetailsStart State.



**NOTE:** To see how the Total\_1 Line and the other Total Lines, to be used as an EndState for the DetailsLine and DetailsStart States, are created, review steps 1 through 8 from the section entitled, "Adding Lines & Parameters in the Text to XML Editor."



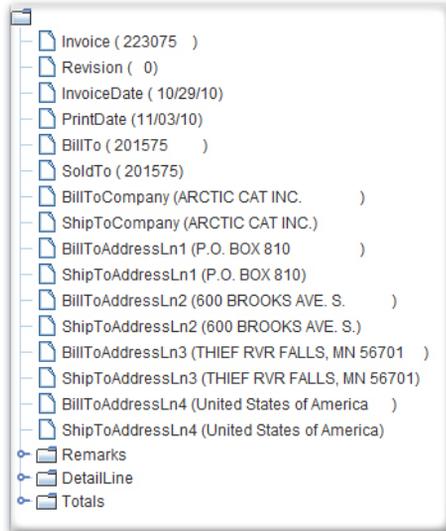
12. Including the Total Lines as States is similar to that of the Line\_30 State. For the EndState of the Total\_1 State, select the FormFeed Line.



13. Click *Test* in the bottom right.



14. The file will be converted to XML. Click *Save* in the bottom right to record the changes.





# *Reference Guide*

---

## State Engine

States read source data line-by-line and can be indefinitely nested. Each Line is checked by the State Engine.

For the Current State, each Nested State is checked to see if it's "Line" value can be matched. If a matching "Line" is found, that State becomes the Current State. If no Nested State matches, each EndState is checked in the same way. If one matches, then the Current State is exited. The default destination is the State in which the Current State is nested. A specific Ancestor State can be referenced, in which case it becomes the Current State. If the referenced EndState definition is not found, the Root State will then become the Current State.

Upon entering a State, if the Line it is based upon has Parameters, the defined Parameters are extracted from the source into the x result.

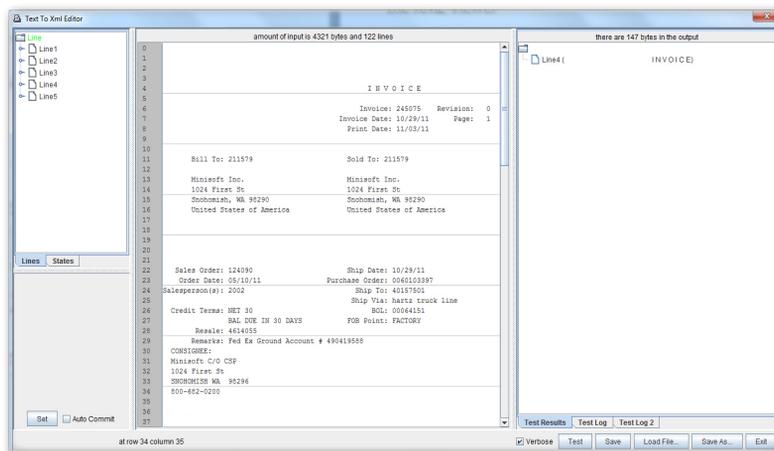
If a State has no Nested States or EndStates, the State exits after extracting data to the parent State. This is the same action as if a single EndState were used with no Line to match or destination defined.

When all the Lines of the source are exhausted, this preprocessor returns XML-formatted data to the Form Template for processing.

# Text to XML Editor Window

The Text to XML Editor has three main window panes:

- ◆ Line and State Viewer
- ◆ Input Data Viewer
- ◆ XML Data Viewer



Each window pane contains different functions and controls to transform a file into XML. The following headings will discuss each window and their options and functions. The Text to XML Editor window contains global options, these options include:

**Verbose:** Optional mode that will display status messages as it is processing. Default is checked.

**Test:** Tests the changes, displaying the XML output, without saving the alterations.

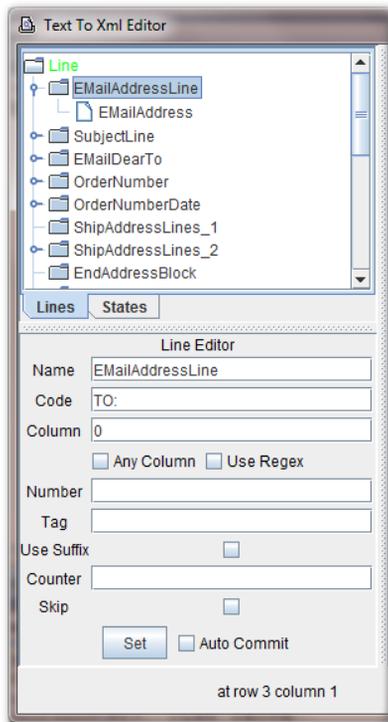
**Save:** Saves the changes and refreshes the XML output.

**Load File:** Allows the user to load a Text to XML conversion file from a directory.

**Save As...:** Opens the Save dialog box, so as to save the Text to XML conversion file under a new name.

**Exit:** Exits out of the Text to XML Editor.

## Line Editor Window and Options



**Name:** An arbitrary character sequence.

**Code:** Character string to match in the data.

**Column:** Column on the line to start matching.

**Any Column:** Match the Code starting in any column.

**Use Regex:** When checked, enter a regular expression for the Code and set Column to 0.

**Number:** Line number, starting at zero, from the top of the input.

**Tag:** Alternative name in the XML output for a line.

**Use Suffix:** Adds a numeric sequence to the line matched.

**Counter:** Name of counter used in place of the default.

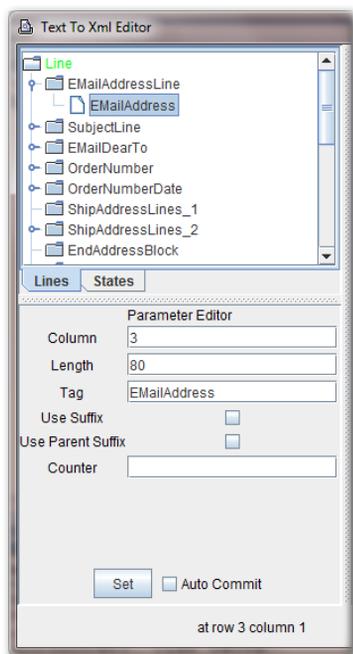
**Skip:** Disables the particular line rule.

**Set:** Sets the changes.

**Auto Commit:** Automatically sets the changes when checked. Default is unchecked.

## Line Editor Parameter Options

When a Line Parameter is selected, the Parameter Editor will display. Parameters are definitions of data to extract. The bottom illustrates the location of the selected Parameter in terms of rows and columns.



**Column:** Where on the line to begin extracting data.

**Length:** Number of characters to extract.

**Tag:** Name of extracted element to be used in the XML output.

**Use Suffix:** Adds a numeric sequence to the line matched.

**Use Parent Suffix:** If a child state, the suffix of its parent will be used.

**Counter:** Disables the particular line rule.

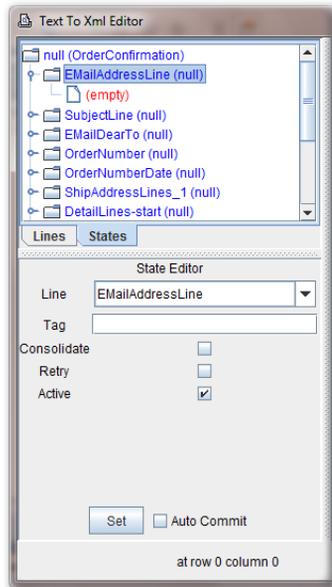
**Set:** Sets the changes.

---

**Auto Commit:** Automatically sets the changes when checked. Default is unchecked.

## State Editor Window

An input file is parsed line by line. States determine when each line is parsed and depending on a State's end state, how often it's parsed. States can be nested.



### State Editor Options

**Line:** The line to match.

**Tag:** An alias name that sets a level in the XML output.

**Consolidate:** When multiple lines match in non-contiguous fashion, it can concatenate all the nodes into one.

**Retry:** Reuse the same input line. This is often used when a line is used as an end state and needs to start a new state.

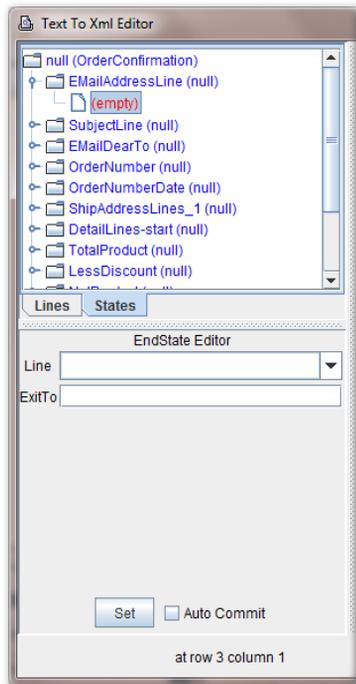
**Active:** Activates or deactivates the state. Default is checked and active.

**Set:** Sets the changes.

**Auto Commit:** Automatically sets the changes when checked. Default is unchecked.

## EndState Options

When an EndState of a State is selected, the EndState Editor will display.



**Line:** Line to drop the user out of the state. Empty means that it leaves the state at the end of the current line.

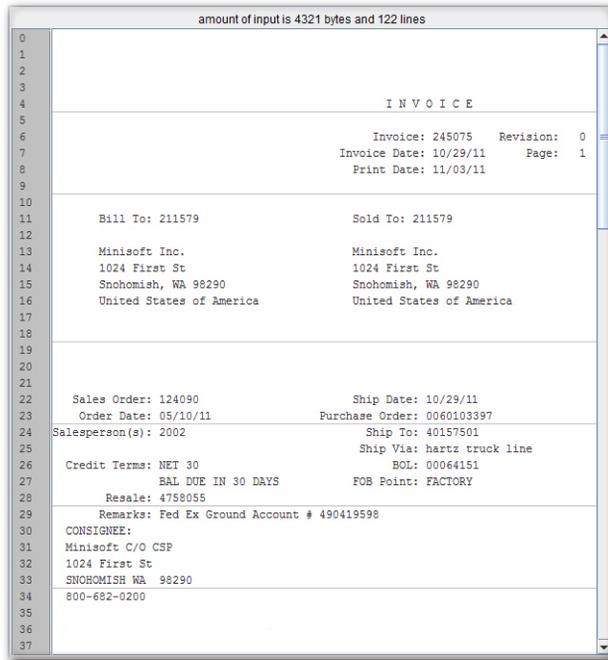
**ExitTo:** Control returns to the named State. If a number is entered, that number of lines is skipped.

**Set:** Sets the changes.

**Auto Commit:** Automatically sets the changes when checked. Default is unchecked.

# The Input Data Viewer

The middle pane of the Text to XML Editor is the Input Data Viewer. The viewer will display the input data as it appears originally from your data application. The viewer is interactive, allowing you to create line numbers, codes, or parameters.



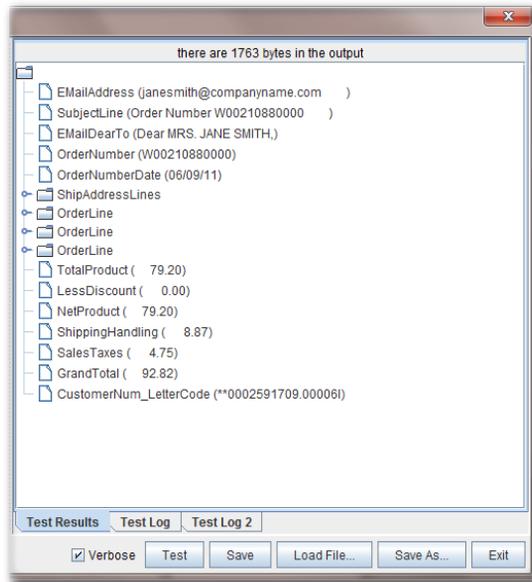
The screenshot shows a window titled "amount of input is 4321 bytes and 122 lines". The window contains a text-based invoice document. The document is displayed in a monospaced font with line numbers on the left side of the window, ranging from 0 to 37. The invoice content is as follows:

```
0
1
2
3
4          I N V O I C E
5
6          Invoice: 245075   Revision: 0
7          Invoice Date: 10/29/11   Page: 1
8          Print Date: 11/03/11
9
10
11         Bill To: 211579           Sold To: 211579
12
13         Minisoft Inc.             Minisoft Inc.
14         1024 First St             1024 First St
15         Snohomish, WA 98290       Snohomish, WA 98290
16         United States of America   United States of America
17
18
19
20
21
22         Sales Order: 124090         Ship Date: 10/29/11
23         Order Date: 05/10/11       Purchase Order: 0060103397
24         Salesperson(s): 2002        Ship To: 40157501
25                                     Ship Via: hartz truck line
26         Credit Terms: NET 30        BOL: 00064151
27                                     BAL DUE IN 30 DAYS   FOB Point: FACTORY
28         Resale: 4758055
29         Remarks: Fed Ex Ground Account # 490419598
30         CONSIGNEE:
31         Minisoft C/O CSP
32         1024 First St
33         SNOHOMISH WA 98290
34         800-682-0200
35
36
37
```

---

# The XML Data Viewer

The right pane of the Text to XML Editor shows the incoming data converted to XML.





# *Glossary*

### *Ancestor State*

Any one of the collection of States from the immediate parent of the Current State to the Root State is an Ancestor State. All States, except the Root State, are nested inside a parent State.

### *Bursting*

Bursting is the process by which a single file is split into multiple, logical files for subsequent processing. Each resultant file can contain one or more pages of Print Output from the source.

### *Current State*

At the start of processing a Line from the source, this is the State that will be referenced. The Current State at the start of processing is the Root State.

### *Form Template*

A file created and edited with the eFORMz composer that contains the definitions and controls used by eFORMz to produce enhanced output.

### *Line*

The definition to which a text or Regex match is made from a line of input data.

### *Matching Line*

If either a case sensitive search of a character sequence or the result from a regular expression (Regex) is true, then there is a Matching Line.

### *Nested State*

Any one of the collection of all States that are part of the Current State that are used in Line-matching is a Nested State.

### *Parameter*

Parameters are definitions of data to extract.

### *Print Output*

The data source that is acted upon. This may or may not have been originally intended for a pre-printed form. While eFORMz can access data in a database, the Print Output is an event that triggers eFORMz to use a Form Template. The Print Output is the usual data source for processing.

### *"Regex"*

Regex is the abbreviated form of regular expression. Regular Expressions are a standard way of finding text using characters sequences. The Text to XML Converter is capable of processing regular expressions, as it can examine text and identify parts that match the provided specifications of a regular expression.

### *Root State*

The base, original state or the State Engine.

### *State*

When a Line match occurs, a State is either entered (State) or exited (EndState).

### *XML*

An XML file is a self-contained file of named elements and values.





1024 First Street  
Snohomish, WA 98290  
U.S.A.

800-682-0200  
360-568-6602  
Fax: 360-568-2923

[sales@minisoft.com](mailto:sales@minisoft.com)  
[support@minisoft.com](mailto:support@minisoft.com)  
[www.minisoft.com](http://www.minisoft.com)



Papiermühleweg 1  
Postfach 107  
Ch-6048 Horw  
Switzerland

+41.41.340.23 20

[info@minisoft.ch](mailto:info@minisoft.ch)  
[www.minisoft.ch](http://www.minisoft.ch)

© 1983 - 2014 by Minisoft, Inc.