
Minisoft® eFORMz™

Reference Manual

Version 10.05

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
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://support.minisoft.com>

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.

©2015 by Minisoft, Inc. Printed in U.S.A.

eFORMz 10.05 - March 2015

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

1. 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.
2. 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.
3. 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.
4. 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.
5. 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.
6. 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.

1. 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.
2. MINISOFT warrants that it has the right to grant the licenses contained in this Agreement.
3. 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.
4. 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.
5. 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. 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

SECTION 1: Menus and Project Properties

Main Menu Options	2
File Menu	2
Project Menu	2
Run Menu	3
Host Tools Menu	4
Window Menu.....	5
Help Menu	5
System Properties.....	6
Viewer Controls	12
Viewer Menu Options.....	13
Variables Viewer	17
Current Project Options	18
Project Properties	19
Page Properties	37
Form Properties	38
Data Suppression Tab	45

SECTION 2: Data Customization

Variables	
Variable Types.....	50
Positional Data Block Variables	50
Node Data Block Variables.....	52
For activeForms Input	54
Queue Monitor Variable.....	55
Global Variables	56
System Variables.....	59

Variable Functions

Concatenate	63
Insert String	67
Set Value.....	72
Setting an Image Value.....	76
Substring	77
Replace	79
Choose One Of.....	83
Set Value from a Declared Constant	84
Set Value from a Keyed Declared Constant	85
Translate	86
Parse Date	87
Right and Left Trim	88
Lowercase and Uppercase.....	90
URL Encoder.....	92
URL Decoder.....	93
UPC-EAN Check Digit	94
Table Lookup	95
SQL Lookup.....	97
Does File Exist.....	99
eVS Barcode Data Generator	100
SmartLabel Design and Barcode	102
Bulk Concatenate	104
Base64 Decode	105
Base64 Encode	105
Regex Replace	106
Modulo	107
HashSHA1	107
String Length	108
String Position	108

Procedures

Calculation.....	110
Set Variable.....	112
Sum	113
Date Calculation.....	113

Sort.....	114
Numeric Format.....	115
Date Format	116
Time Interval Format	117
SQL Load.....	118
SQL Update	119
SQL End Transaction.....	121
Load Variable from a Declared Constant	122
Load Variable from a Keyed Declared Constant.....	123
Load Variable from a Variable.....	124
Load Variable from Project Output.....	124
Load Variable from XML Node	125
Load a Component.....	126
Call a Web Service	127
Call a Dynamic Service	128
File List	129
Printer List	130
Rule Actions	
Place Text Action.....	132
Page Selection	135
Text Rotation	136
Text Word Wrapping.....	137
Place Text Direct Reference.....	138
Load Image	140
Load Image Based on a Condition.....	143
Page Number.....	145
Translate.....	146
Notify	148
Output from Project.....	150
Constant Actions	
Creating a Constant	152
Image	153
Text.....	154
1D Barcode	156

2D Barcode	158
Special Fonts	159
Line.....	160
Box	161
Conditional Processing	
Adding a Condition	163

Appendix A: Advanced Processing

Command File Processor	168
Script Example.....	168
Parameters	168
Preprocessor Data Parsing Control Codes.....	169
eSUBMIT Command Processing	169
Overview	169
Details	169
Commands	170
Quick Reference Guide	174
User Configuration Files.....	176
Custom Form Sizes.....	177
Preprocessors and External Functions	177
Font Configuration	178

Appendix B: Database URL and Driver Classes

Supported URL and Driver Classes	180
DB2 (iSeries)	180
Image and Eloquence.....	180
Informix	181
Microsoft Access	181
MySQL.....	182
Oracle	183

Progress.....	184
SQL Server	184
Sybase	185

Appendix C: OnDemand Print Utility

OnDemand Print.....	187
---------------------	-----

Index

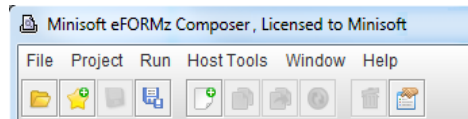
SECTION 1

Menus and Project Properties

Main Menu Options

The Composer menu bar has the following five menus:

File, Project, Run, Host Tools, Window, and Help.



File Menu

The File Menu controls the project file with the following items:

Open Project

Opens any previously saved project template.

New Project

Creates a new project template with the supplied name.

Save Project

Saves the currently loaded project by its current name.

Save Project As

Saves the currently loaded project by a new user supplied name.

System Properties

Controls the operation of the eFORMz Composer.

Exit

Closes the currently loaded project and exits the Composer.

Project Menu

The Project Menu controls project items with the following options:

Add New Page to Project...

Adds a new page to your project. You will need to add a page for each logical page of data within your report. Most projects will only contain one logical page regardless of how many physical pages are contained within the report.

Add Form to Page...

Adds a form to a logical page. Adding multiple copies of the same form causes a report to print multiple copies of a page of data.

Replace Form

Replaces an existing form with another. Used extensively when making alterations to your form.

Reload Form

Reloads a previously imported RTF document automatically.

Remove Selected Item

Removes a selected page, form, or rule from the current project. A warning message will be issued prior to the delete.

Properties...

Displays the Properties dialog box of the selected item in the Current Project window (Project, Page, or Form).

Run Menu

Print

Sends a merged print file of your form and current data file to a selected printer.

Output..

PDF File

Produces a merged pdf file of your form and data output.

eFORMz Document Files

Produces an .efd file for eReader.

PCL Files (B/W)

Produces a merged file of your form and data in black/white pcl format.

PRN Files (B/W)

Produces a merged file of your form and data in black/white pcl format.

PCL Files (Color)

Produces a merged file of your form and data in color pcl format.

PostScript Files

Produces a merged file in the standard format for a Postscript printer.

SECTION 1

XML

Produces a merged file in standard XML format for a Zebra printer.

HTML

Produces an HTML file of your output.

TEXT

Produces a print file in standard text format.

AFPDS

Produces an AFPDS print file.

ZPLII

Produces a print file in Zebra's ZPLII format.

TIFF, JPEG, PNG, GIF, BMP

Produces an image file in one of the above formats.

Process w/eDIRECT or eDIRECT PLUS

Processes output through eDIRECT.

Host Tools Menu

Local Director Toolkit

Provides a GUI interface for configuring the eFORMz job process on a Windows operating system (Vista/7/Server 2008/8). For further information, see the *eFORMz Director Manual*.

Install an extension

An eFORMz Extension is a file set that can be installed within an installed copy of eFORMz that may be supplied to a customer by a Minisoft support representative. The extension can include, files, directories and XML file modifications. These items are scripted through an XML file.

AS400

Provides a GUI interface for configuring the eFORMz job process on your AS400. For further information, see the *eFORMz Director Manual*.

Unix

Provides a GUI interface for installation, saving, uploading, and downloading files from your Unix system. For further information, see the *eFORMz Director Manual*.

Window Menu

The *Window* menu allows you to select which windows you would like to have active within the eFORMz Composer.

Project

(default enabled) Displays the *Current Project* window.

Viewer

(default enabled) Displays the form and data *Viewer* window.

Variables

(default disabled) Displays the *Variables* viewer window.

Reset

Restores the default window settings.

Text to XML Editor

Works in conjunction with the *Project Properties > Advanced Preprocessor > Text to XML Converter*.

Help Menu

eFORMz Manual

Provides quick access to the eFORMz pdf manual.

About Minisoft eFORMz

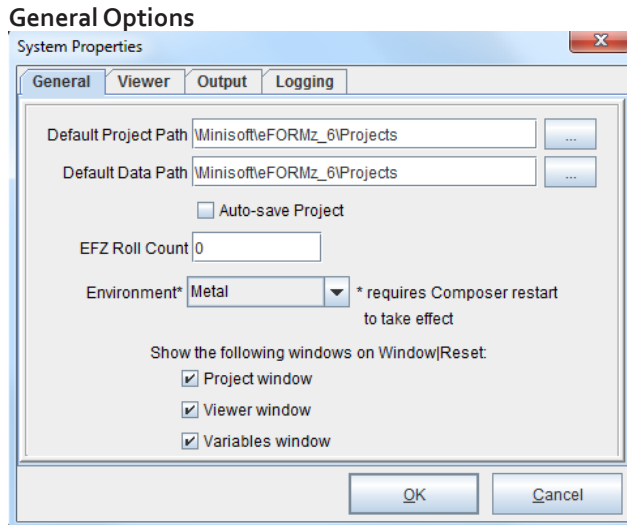
Displays the eFORMz copyright date and version number. Provides an activation license check.

Available Character Sets

Contains a list of accessible character sets.

System Properties

The eFORMz *System Properties* sets the settings for all projects. To access the eFORMz System Properties, select *System Properties* from the eFORMz *File Menu*.



Default Project Path

Sets the default folder that will be opened when retrieving a project file.

Default Data Path

Sets the default folder that will be opened when retrieving a data file.

Auto-save Project

Automatically saves changes made to your project file.

EFZ Roll Count

Permits the automated backup of projects based on a preset counter. For using the *EFZ Roll Count*, consult the following tutorial:

<http://support.minisoft.com/index.php/revision-handling/>

Environment

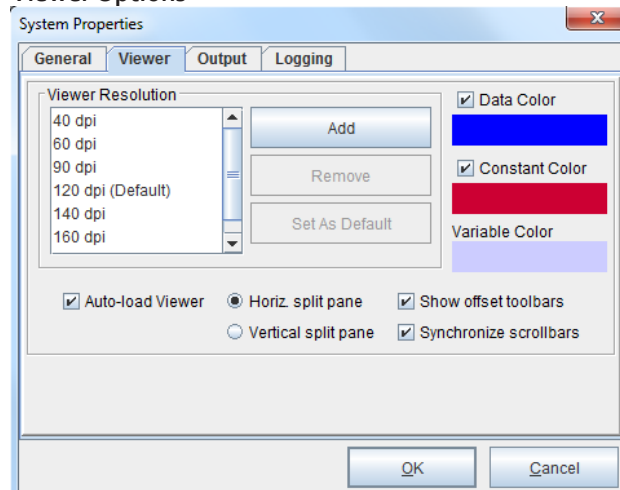
Changes the appearance of the eFORMz application. Shutting down the eFORMz application is required for an appearance to change. Options include: *Metal*, *CDE/Motif*, *Windows Classic*. Default is *Metal*.

Window|Reset

When checked, the associated window will appear if the Window Reset is selected from the Composer Task bar.

- Project
- Viewer
- Variables

Viewer Options



Viewer Resolution

(Default 120dpi) Controls the default zoom levels. Customize your viewer controls by clicking *Add* or *Remove*.

Data Color

Allows you to change the data color displayed in the viewers.

Constant Color

Allows you to change the constant color displayed in the Form and Data Viewer.

SECTION 1

Variable Color

Allows you to change the variable color displayed in the Data and Form Viewer.

Auto-load Viewer

When checked, the Form and Data Viewer automatically reloads when a change has been made to the project rules, properties, or form.

Horizontal Split Pane

Allows you to view your input and output data simultaneously using the horizontal split pane button. Default is checked.

Vertical Split Pane

Allows you to view your input and output data simultaneously using the vertical split pane button.

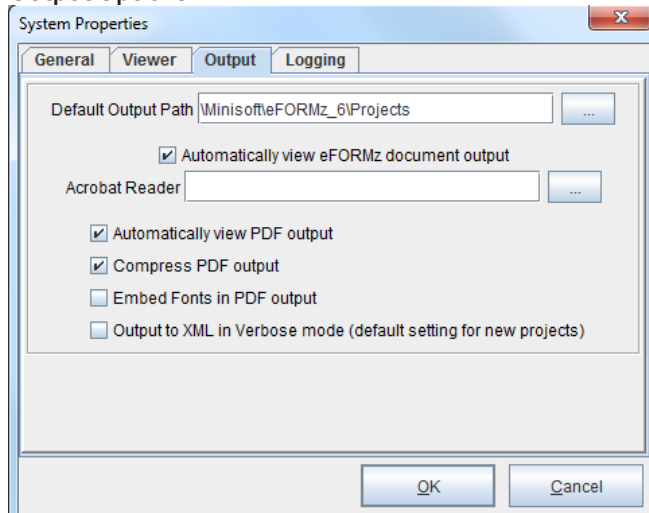
Show Offset Toolbars

When checked, the Form and Data Toolbar displays allowing you to easily adjust the positioning of your form and data. Default is checked.

Synchronized Scrollbars

Allows scroll bars within the Viewer to become synchronized. Default is checked.

Output Options



Menus and Project Properties

Default Output Path

Specifies the folder the output will be directed to.

Automatically view eFORMz document output

When checked, automatically starts the eReader when an .efd file is created.

Acrobat Reader

Path to the Adobe Acrobat reader executable. Specify if you wish to use a reader other than system default.

Automatically View PDF Output

When checked, the PDF file reader will automatically start after a PDF file has been built from eFORMz. *Note:* Path must be provided for Acrobat Reader.

Compress PDF Output

Compresses the eFORMz pdf file output allowing smaller file sizes.

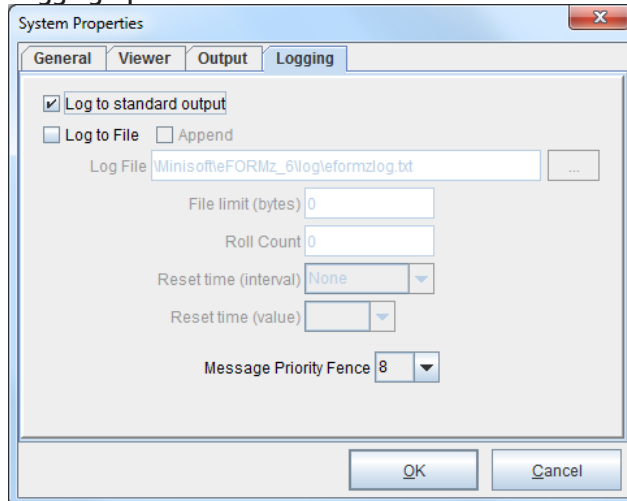
Embed Fonts in PDF Output

Embedded fonts prevents font substitution when a pdf file is viewed or printed. Text will remain in its original font.

Output to XML in Verbose mode

Default unchecked. When unchecked the XML output is condensed using token references. Checked will cause the XML output to be represented in full text XML verbiage.

Logging Options



Log to Standard Output

When checked, logs all messages and standard output from eFORMz to a DOS window.

Log to File

When checked logs all messages and standard output from eFORMz to a text file.

Log file

When "Log to File" is checked use the *Log File* field to specify the path and file name of the log file.

File limit (bytes)

Size in bytes of the log file where a new write will result in rolling. The action of closing the current log file and opening a new log file is called "rolling" the log file.

Roll Count

Limits the number of retained log files.

Reset time (interval)

The type of time interval that the *reset time (value)* uses. Options include: *Years, Months, Days and Hours*. The default value is *None*.

Reset time (value)

The numeric value of reset time intervals to allow for the time a log file to reach the maximum age before rolling. Options include: *1-5 Years*, *1-12 Months*, *1-30 Days* and *1-24 Hours*.

Append

Permits log messages to be appended to the log file.

Message Priority Fence

A message priority fence can filter what messages are displayed or written to the log file.

- 0-3 - All informative messages
- 4-7 - Unused
- 8-11 - Status messages (i.e.: pages output, etc...) only
- 12-15 - Error and other high priority messages only

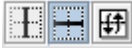
Minisoft recommends a log priority of 8. If the log priority is set below 8, it could adversely affect performance.

Viewer Controls



Zoom Control

Controls the zoom level for viewing. The view setting will display to the right of your zoom controller icons. To personalize your view settings, from the eFORMz Composer File Menu select *System Properties: Viewer tab > Zoom Control*.



Vertical/Horizontal Panes

Allows you to customize your Viewer panes to view horizontally or vertically your input and output data simultaneously. Use the *Flip Pane* button to switch the Viewer windows. If you wish to view only one pane, slide the pane separator to one side.



Color Control

Enables/Disables color display in the Input Data Viewer and Form and Data Viewer.



Drag Control

Allows you to lock down movable elements in the Form and Data Viewer. Default is unlocked.



Previous/Next Buttons

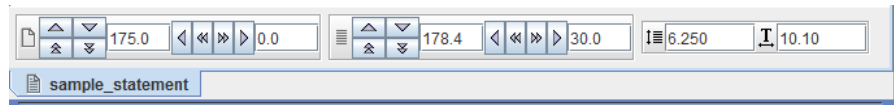
Allows you to move back and forth between data pages.

3408.0 : 1040.0

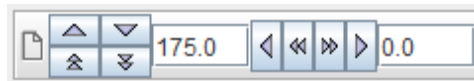
Row/Column Coordinates

The position of your mouse in the Form and Data Viewer. Coordinates based in de-cipoints.

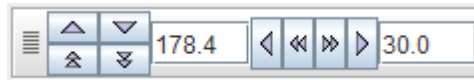
Offset Adjustment Toolbars



When checked, the Form and Data Toolbars displays allowing you to easily adjust the positioning of your form and data.



Form Offset Adjustment Toolbar



Data Offset Adjustment Toolbar



Lines Per Inch - The number of data lines in every vertical inch.



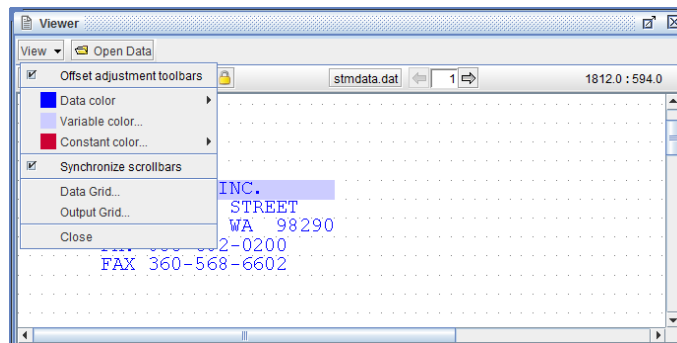
Characters Per Inch - The number of characters per inch horizontally. Used in conjunction with Fixed fonts.



Height - The vertical height of each data character. Used in conjunction with Proportional fonts.

Viewer Menu Options

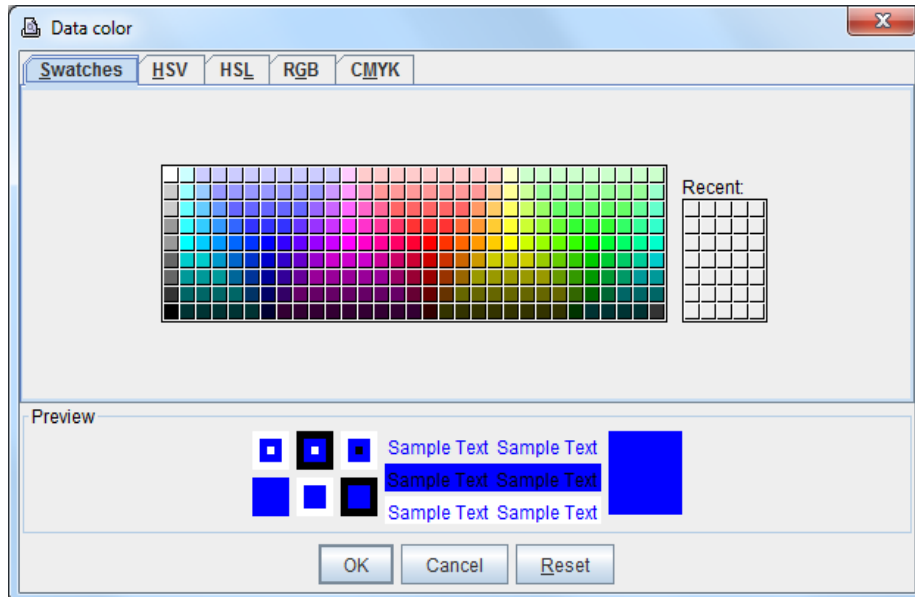
The eFORMz Viewer has a main menu option that allows you to control different aspects of the viewer such as the data grid, scrollbars, and viewer colors. Note some of the options available in the following headings.



SECTION 1

Data, Variable, and Constant Color

Allows you to change the data color displayed in the Viewer.



Data Color

Allows you to change the data color displayed in the Viewer. Disable deactivates the color function. Enable is the default.

Variable Color

Allows you to change the variable colors displayed in the Viewer.

Constant Color

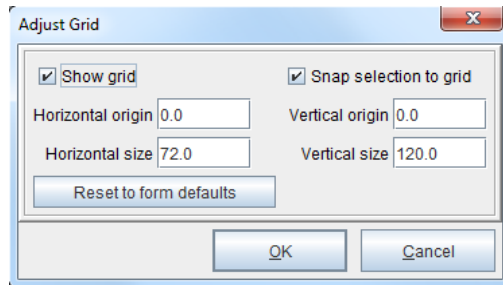
Allows you to change the constant colors displayed in the Viewer. Disable deactivates the color function. Enable is the default.

Synchronize Scrollbars

When checked, scroll bars become synchronized in the Viewer windows.

Data Grid

Allows you to include/exclude the data grid from the Input Data Viewer. Customize the data grid size using the horizontal and vertical decipoint values.



Options Available:

Show Grid

Displays the data grid for easy character selection.

Snap Selection to Grid

Allows you to precisely align objects or data to the grid.

Horizontal/Vertical Origin

Sets the horizontal and/or vertical start of the grid. Values are in decipoints.

Horizontal/Vertical Size

Sets the horizontal and/or vertical size between each line of the grid. Values are in decipoints.

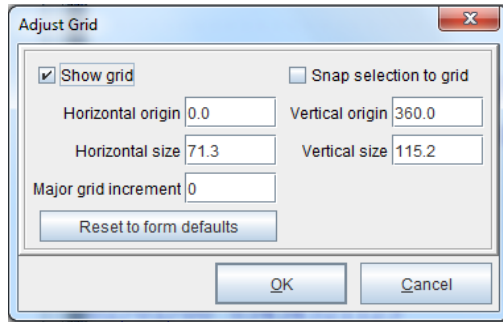
Reset to Form Defaults

Sets origin of size to the default of the form.

SECTION 1

Output Grid

Allows you to include/exclude the output grid from the Form and Data Viewer. Customize the data grid size using the horizontal and vertical decipoint values.



Options Available:

Show Grid

Displays the data grid for easy character selection.

Snap Selection to Grid

Allows you to precisely align objects or data to the grid.

Horizontal/Vertical Origin

Sets the horizontal and/or vertical start of the grid. Values are in decipoints.

Horizontal/Vertical Size

Sets the horizontal and/or vertical size between each line of the grid. Values are in decipoints.

Major Grid Increment

Controls the number of large increments contained in the grid.

Reset to Form Defaults

Sets origin of size to the default of the form.

Close Option

Closes the currently displayed tab in the Viewer.

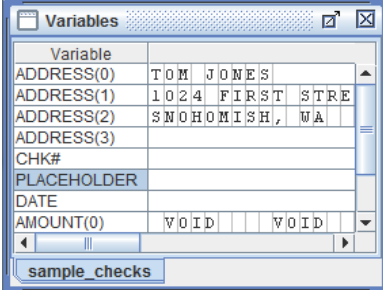
Variables Viewer

The eFORMz Variables Viewer window displays the details of each variable. The information included pertains to:

- ◆ The name of the variable.
- ◆ The data contained within the variable.

The *Variable* column shows the name of the variable and the number of times it is repeated.

The column to the right presents the data that the variable contains. Variables repeated more than once occupy the corresponding number of lines.



The screenshot shows a window titled "Variables" with a table of data. The table has two columns: "Variable" and a data column. The data is as follows:

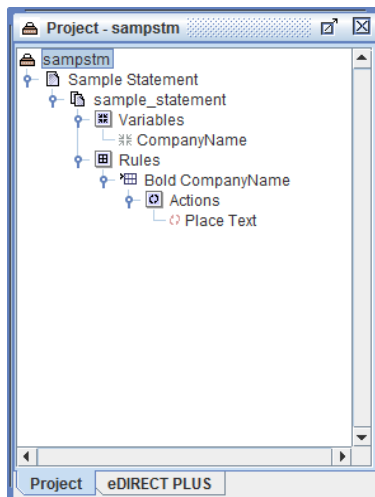
Variable	
ADDRESS(0)	TOM JONES
ADDRESS(1)	1024 FIRST STRE
ADDRESS(2)	SNOHOMISH, WA
ADDRESS(3)	
CHK#	
PLACEHOLDER	
DATE	
AMOUNT(0)	VOID VOID

At the bottom of the window, there is a text field containing "sample_checks".

Current Project Options

The Current Project window displays the entire project layout:

- ◆ Project Name
- ◆ Global Variables
- ◆ Page(s)
- ◆ Form(s)
- ◆ Input and Constant Fields
- ◆ Variables
- ◆ Procedures
- ◆ Form and Rule Conditions
- ◆ Document Template
- ◆ Rules
- ◆ Actions

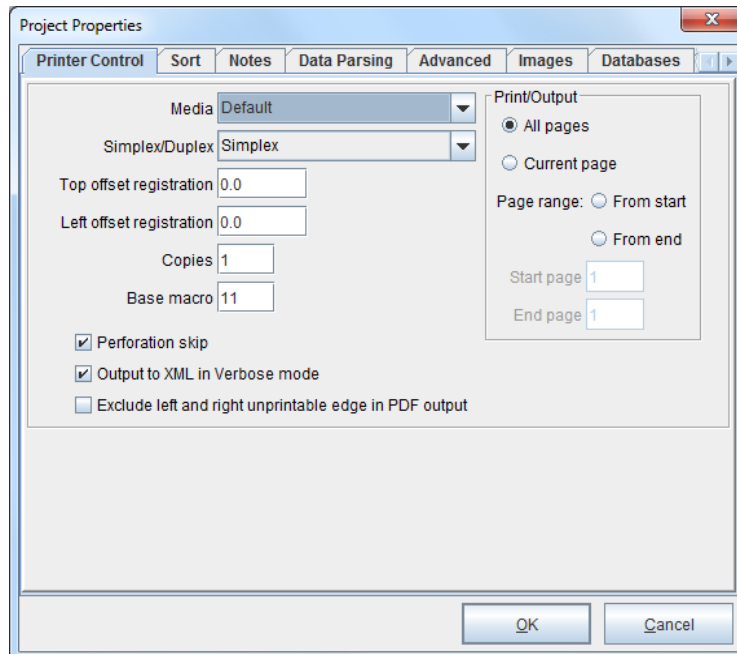


Project Properties

The *Project Properties* controls a project as a whole, for example:

- ◆ print orientation
- ◆ number of each individual copies
- ◆ and data parsing.

To access the Project Properties, select the project name from the Current Project window, *right click*, and select *Properties*:



Printer Control Options

Media

Used to put logical output page(s) onto a physical page. Uses would be to arrange logical pages n-up (2-up) on a physical page or to rotate a logical page on a physical page. When "Media" is set to "Default" the physical page is the same as the logical page. "Media" entries are defined in eFORMzCfg.xml.

Simplex/Duplex

Simplex
Long-Edge Duplex
Short-Edge Duplex

Top Offset Registration

Sets the top margin position for printing (in decipoints).

Left Offset Registration

Sets the left margin position for printing (in decipoints).

Copies

Sets the number of copies printed for each physical page.

Base Macro

Specifies the macro ID number sent to the printer. Use when a printer contains saved macro's in print memory. Default is 11.

Perforation Skip

A setting often used with tractor-fed printers. Unchecked allows you to use the maximum number of lines per page. Checked reduces the number of available printed lines. Default is checked.

Output to XML Verbose Mode

Default unchecked. When unchecked the XML output is condensed using token references. Checked will cause the XML output to be represented in full text XML verbiage.

Exclude left and right unprintable edge in PDF output

Excludes the left and right margin from pdf file output.

Menus and Project Properties

Print/Output - All Pages

When specified, the merge process will print all pages within a data file.

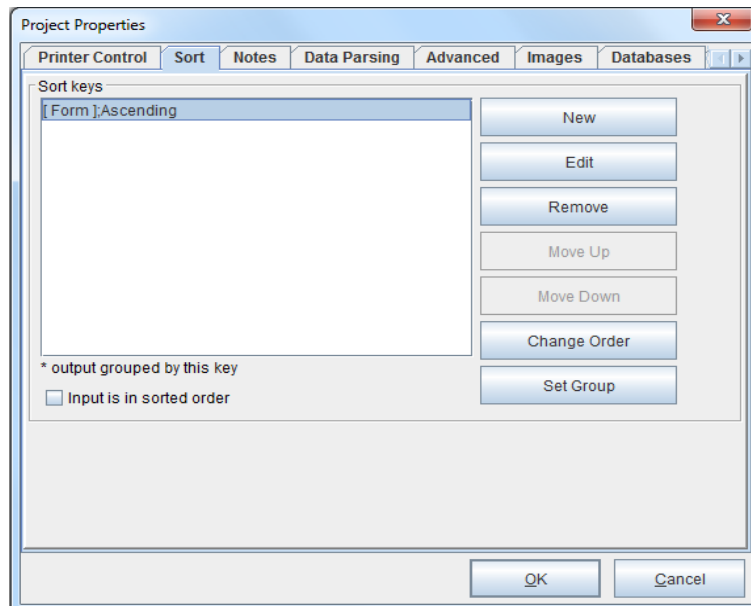
Print/Output - Current Pages

When specified, the current page being viewed within the eFORMz Viewer will be printed.

Print/Output - Range of Pages

When selected, the Start Page and End Page fields will become available. Enter the range of pages you would like to print.

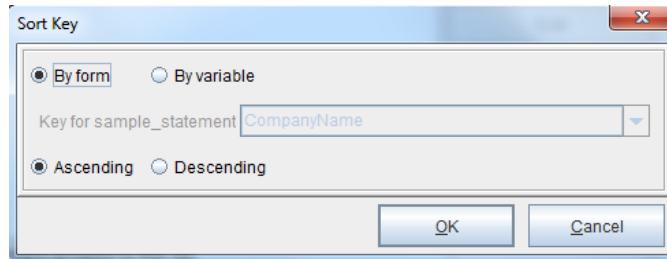
Sort Options



SECTION 1

The *Sort* tab allows you to sort your data file based on a variable or form. To sort:

1. Select *New* and choose the variable or form you would like to sort by as well as the Sort Order. Then click *OK*.



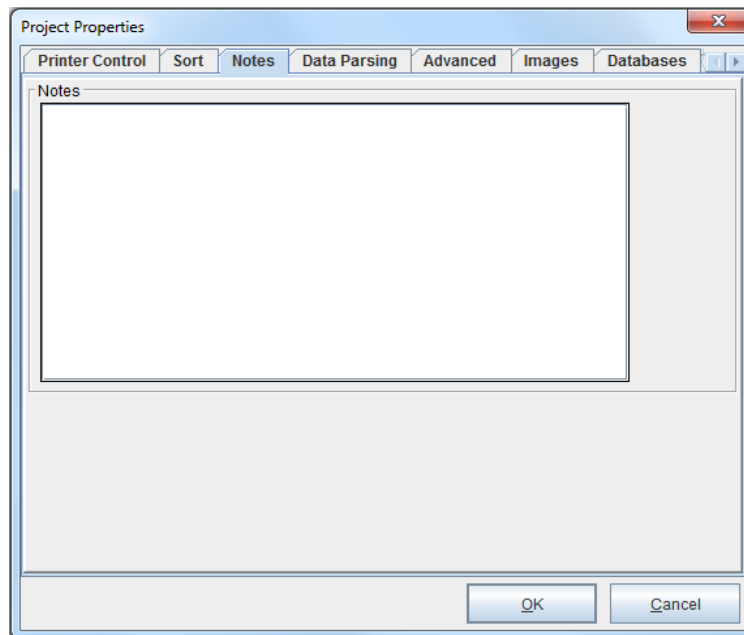
2. The variable or form will appear in the 'Sort Keys' field.
3. To change the sort order highlight the variable or form and select *Change Order* (options: *Ascending* or *Descending*). The Set Group option will arrange the order sequence of multiple keys when a particular key is selected. The Input is in Sorted Order.
4. Use the *Remove* button to remove any variable or form from the 'Sort Keys' field.
5. The Move Up and Move Down options will allow you to change the 'Sort Keys' order.

Note: Multiple variables or forms can be added.

Cross-Reference: For sorting pages by data, view the following tutorial:

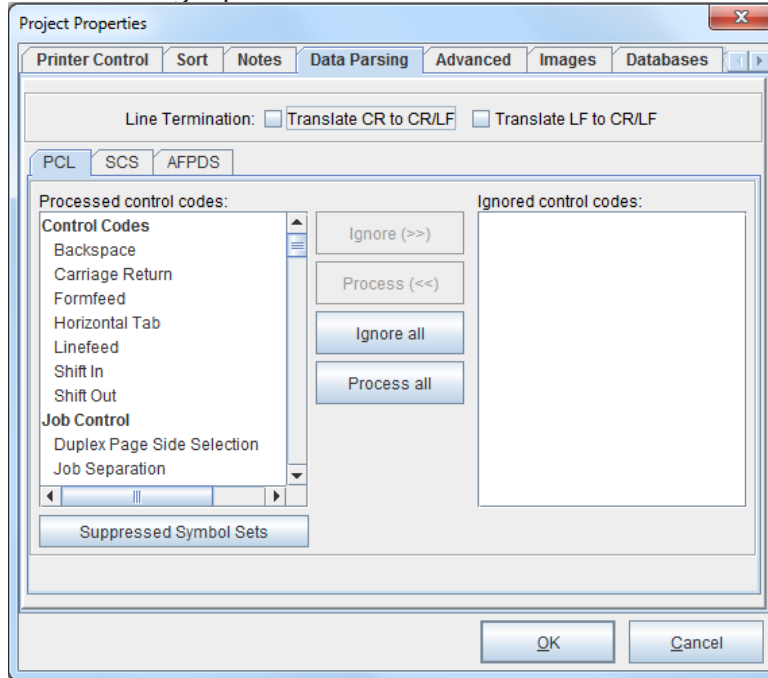
<http://support.minisoft.com/index.php/adding-page-sort-to-an-eformz-project-video-only/>

Notes Options



This section is available for documenting changes and providing a general explanation of your process.

Data Parsing Options

**Line Termination:****Translate CR to CR/LF**

When selected, a data file containing only carriage returns is translated into a data file containing carriage returns and line feeds.

Translate LF to CR/LF

When selected, a data file containing only line feeds is translated into a data file containing carriage returns and line feeds.

Print Streams:**PCL**

For use with a Printer Command Language data stream.

SCS

For use with a System Network Architecture Character Stream, which contains SNA character string controls.

AFPDS

For use with an Advanced Function Printing Data Stream.

Processed Control Codes

Contains a listing of possible control codes in your data file.

Ignore (>>)

Moves a selected control code to the Ignore Control Codes list.

Process (<<)

Moves selected ignored control code to the Process Control Codes list.

Ignore all

Moves all control codes listed to the Ignore list.

Process all

Moves all ignored control codes to the Process Control Codes list.

Ignored Control Codes

Contains a listing of all ignored control codes.

Suppressed Symbol Sets

(Advanced feature) Allows Softfont symbol sets to be suppressed from the incoming datastream.

Advanced Options

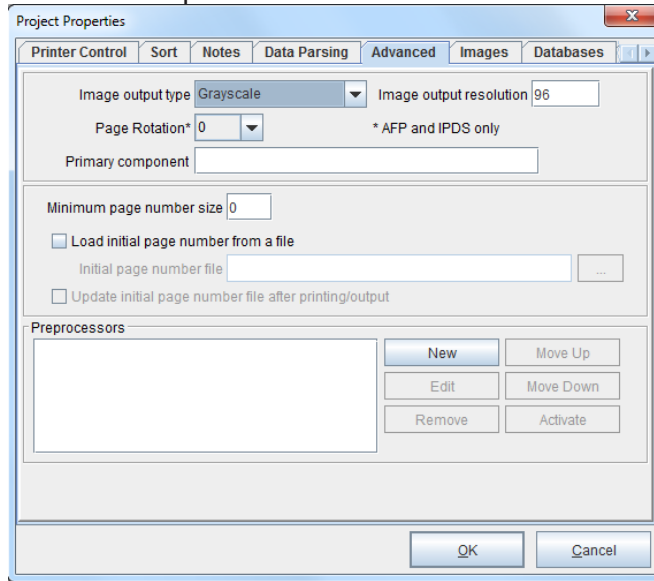


Image output type

When outputting an image, select from Grayscale, Black and White and Color options for the image output type.

Image output resolution

The image resolution. Default is 96.

Page Rotation

Positions the page by degrees for Advanced Function Printing and Intelligent Printer Data Stream applications only.

Primary component

For use with a composite data file. The <Component> tag is specified here in conjunction with the Output from Project Rule Action.

Minimum page number size

Minimum number of characters for page number. If the number of digits of the current page number is less than this value then it will be padded with leading zeroes up to the minimum character size.

Load initial page number from a file

Loads a page number from an external file.

Initial page number file

Specified path to the external page number file.

Update initial page number file after printing/output

Updates the page number file automatically after each process.

Preprocessors

Parses incoming data in one of the following ways:

Macro Insert - Permits the insertion of a macro into the file.

Skip until a string is found - Ignores incoming string commands until the specified string is found. Specify the number of occurrences the specified string should be skipped.

Replace occurrences of a string - Replaces a specified string with another.

Multiple replace occurrences of a string - Replaces multiple strings with another.

Command parsing - For custom command parsing.

Sequential Line Marker - Marks incoming data based on a search criteria.

Text file pagination - Searches for a string in a particular column and then issues a formfeed (\f) at the beginning of that line.

Flat file parser -

Ability to parse as input comma delimited or EDI formatted files.

Text to XML Converter -

Converts text files to XML.

Note: For Command File Processor Parameters and Preprocessor Data Parsing Control Codes, see *Appendix A: eFORMz Preprocessor*. For more on the Text to XML Converter, view the *Text to XML Manual*.

SECTION 1

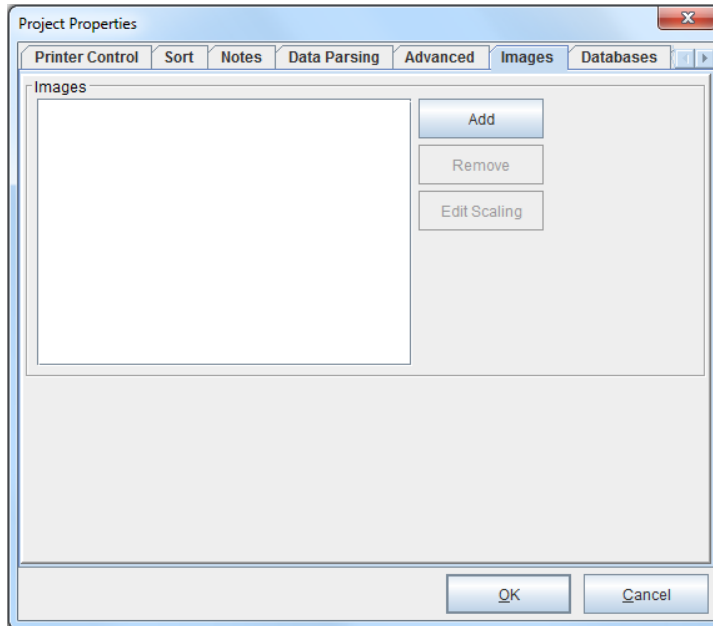
SQL ResultSet - The result set of an SQL statement is passed in XML format to the next preprocessor or to the main project.

Project - Processes the data file with a project. The output of the project is passed in efd format to the next preprocessor or to the main project.

Character Set Translator - Change from one character set to another.

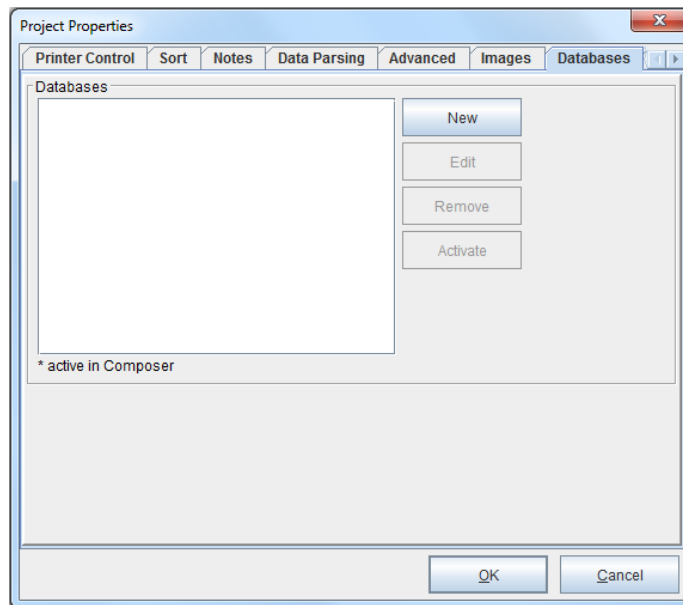
Insert a string - Allows the insertion of a string at a particular byte location.

Images



The Images dialog box allows you to import images (jpg, gif, bmp, tiff, pdf, prn) into your project file to be used later in an action or variable. Note that all images added to a project are saved in the output in case they are referenced on any of the pages.

Databases Option



New

Allows you to configure and maintain database connections.

Note: The Driver Class and URL fields are case sensitive. To save the connection information select *Save to File* to create a text file of your connection information. When using the same database for a different project, click the *Load from File* button to select the text file that was created earlier.

If the database cannot be read, a qualifying record cannot be found, or the value of the column is NULL then the variable value will be set to the empty string.

For supported database driver and url path specifications, see *Appendix B: Database URL and Driver Classes*.

Edit

Permits modifications to a database in the list.

Remove

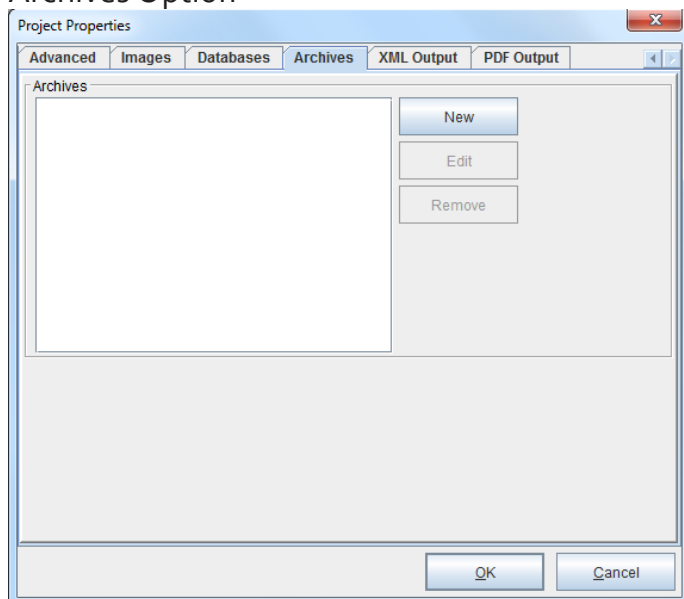
Deletes a database entry from the list.

SECTION 1

Activate / Deactivate

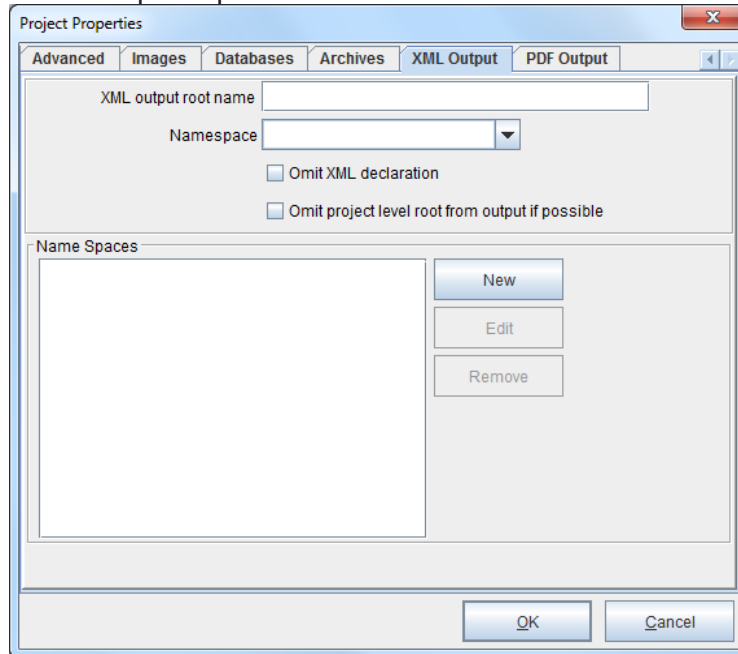
Enable or disable connectivity to the selected database when viewed from the Composer only.

Archives Option



The Archives tab works in conjunction with eDIRECT. The Archive can be configured from within the project or externally in the Archives.xml file. This configuration file consists of a name and a provider name and any other parameters specific to the provider.

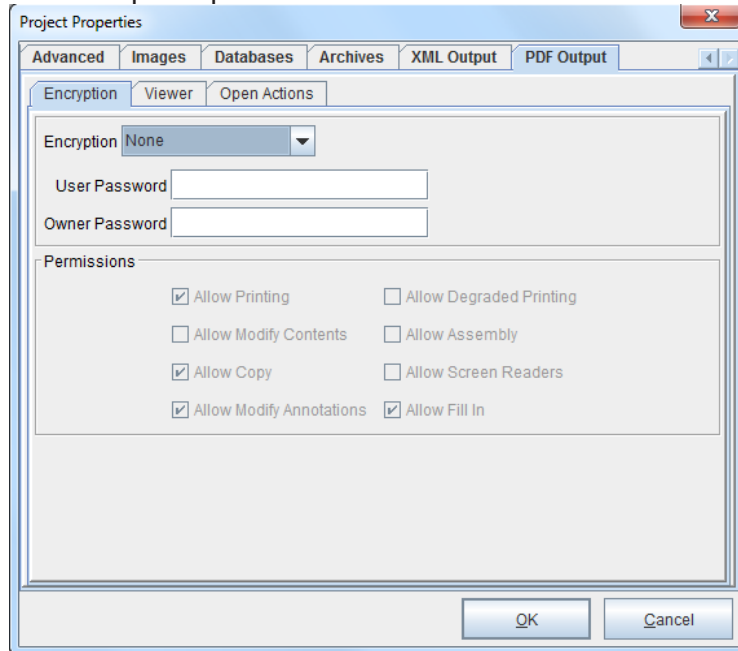
XML Output Option



The XML Output tab controls the name and namespace of a root element when doing XML output. It controls whether the "<xml?" declaration is included. It also allows the definition of namespaces. The namespaces can be specified on the root and any record elements.

Cross-Reference: For more on generating XML output with eFORMz, consult the *Record Element Mini-Manual* available from the Minisoft Support Wiki.

PDF Output Option



Encryption

Encrypt your PDF documents in one of the following ways:

Encryption

Options include Standard 40-bit, Standard 128-bit and AES 128-bit encryption. Encryption is used to enforce restrictions on your document. The permissions that can be granted or restricted depend on the strength of the encryption (40-bit vs. 128-bit). If you are using 40-bit encryption, every permission that has the remark 128-bit below is granted automatically. If you want to revoke these permissions, you need to use 128-bit encryption.

User Password

Refers to the 'open password'.

Owner Password

Refers to the 'permissions password'.

The maximum password length is 32 characters. You may enter 32 characters or more but only the first 32 characters will be taken into account. One or both of the passwords fields can be empty. If a user password is not specified, all users will be able to open the document without being prompted for a password, but the permissions and restrictions (if any) will remain in place.

For a higher level of encryption for your document, choose 128-bit key length and always set both passwords, using different strings and all 32 characters for each one. If a password you choose is shorter than 32 characters, it will be padded with default padding.

Allow Printing

(Default checked) Printing the document.

Allow Degraded Printing

(128-bit) Printing the document, lesser quality printout than *Allow Printing*.

Allow Modify Contents

For example, changing page content, inserting or removing pages.

Allow Assembly

(128-bit) Inserting, removing, rotating pages and adding bookmarks are allowed. Contents of a page cannot be changed, unless *All Modify Content* is allowed.

Allow Copy

(Default checked) Copying or extracting text and graphics from the document, including technologies such as screen readers or other accessibility devices.

Allow Screen Readers

(128-bit) Extracting text and graphics from document using an accessibility devices.

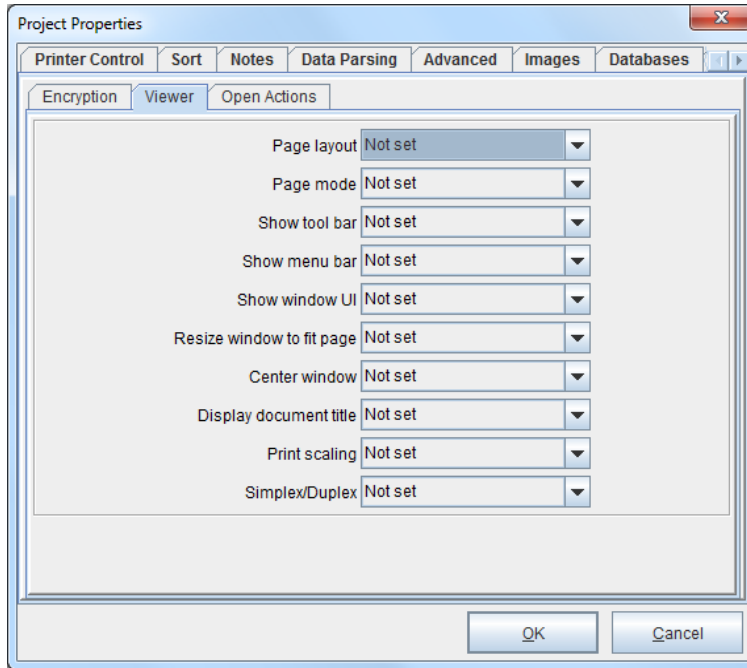
Allow Modify Annotations

(Default checked) Adding or modifying text annotations and interactive form fields.

SECTION 1

Allow Fill In

(Default checked) (128-bit) Filling form fields; adding or modifying annotations only if *Allow Modify Annotations* is granted.



Viewer

Adjust the options for viewing the PDF document:

Page layout

The page layout to be used when the document is opened. Options include:

Single page - Display one page at a time (default).

One column - Display the pages in one column.

Two columns, odd on the left - Display the pages in two columns, with odd-numbered pages on the left.

Two columns, odd on the right - Display the pages in two columns, with odd-numbered pages on the right.

Two pages, odd on the left - Display the pages two at a time, with odd-numbered pages on the left.

Two pages, odd on the right - Display the pages two at a time, with odd-numbered pages on the right.

Page mode

Sets how the document should be displayed when opened. Options include:

None - Neither document outline, nor thumbnail images visible (default).

Show Outlines - Document outline visible.

Show Thumbnails - Thumbnail images visible.

Full screen - Full-screen mode. No menu bar, window controls, or other windows will be visible.

Show tool bar

Displays (or hides) the toolbar.

Show menu bar

Displays (or hides) the menu bar.

Show window UI

Displays (or hides) the window user interface.

Resize window to fit page

Adjusts the window to fit the page.

SECTION 1

Center window

Centers the window.

Display document title

Displays the title of the PDF document.

Print scaling

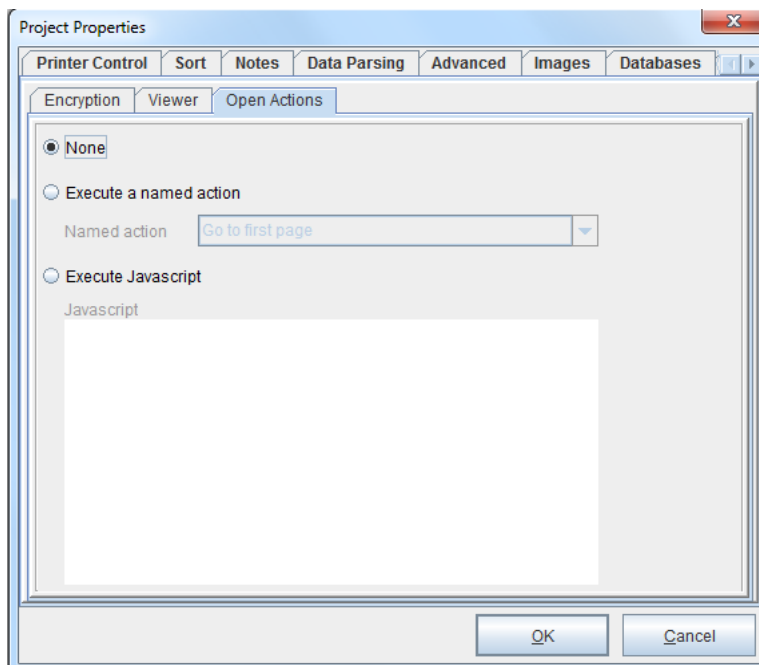
The Default option scales the PDF to a printable area.

Simplex/Duplex

(Default checked) Adding or modifying text annotations and interactive form fields.

Allow Fill In

Printing options include Simplex, Short Edge Duplex, and Long Edge Duplex.



Open Actions

Set an action to be performed at the opening of the document.

None

(Default) No action set.

Execute a named action

Options include:

Open print dialog - Opens the print dialog upon opening the PDF.

Go to first page - Displays the first page of the PDF upon opening.

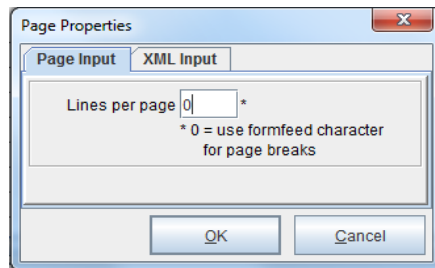
Go to last page - Displays the last page of the PDF upon opening.

Execute Javascript

Enter a Javascript statement to execute an action.

Page Properties

The *Page Properties* dialog box provides general information about your logical page. To open the Page Properties dialog box, select a project page from the Current Project window, *right click*, and select *Properties*:



Page Input:

Lines Per Page

Number of lines per each data page. Default settings:

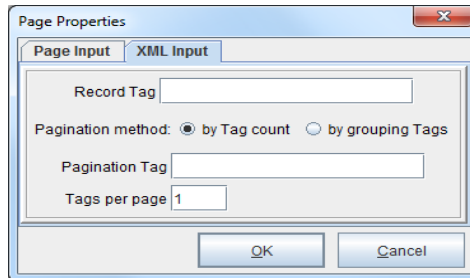
Portrait = 66 lpi

Landscape = 51 lpi

*0=Use formfeed character for page breaks

Zero indicates that the data contains formfeeds. Page breaks will then be issued when a form feed appears in the merge process.

SECTION 1



XML Input

Define an XML data source.

Record Tag

The *Record Tag* is the name of the tag in the XML data file that represents a new data page. The name is an absolute node name. If left blank the record tag will be the root.

Pagination method

Define how pages are set. Default is by *Tag count*.

Pagination Tag and Tags Per Page

The *Pagination Tag* is the name of the tag to use for pagination. It is relative to the *Record Tag*. Multiple occurrences of the named tag within the tag named by the *Record Tag* will appear on separate data pages based upon the value of *Tags per page*. If *Pagination Tag* is blank, no pagination takes place.

Form Properties

The *Form Properties* controls the placement of the form and data. To access the *Form Properties*:

1. From the Current Project window select a project form, highlight *Properties* and select *General*.
2. The *Form Properties* dialog box will display with multiple tabs including *General*, *Font*, *Data Relocation*, *Data Suppression*, *Zebra XML*, *HTML Output* and *Test Data*.

General Tab

The screenshot shows the 'Form Properties' dialog box with the 'General' tab selected. The 'Page Size' is set to 'Letter', 'Orientation' to 'Portrait', and 'Lines per inch' to '6.250'. The 'Top offset' is '175.0' and 'Left offset' is '0.0'. The 'Data' column has '178.4' and '30.0' values. Under 'Data viewer size', 'Match page size' is selected. There are checkboxes for 'Exclude data' and 'Overlay on previous form'. The 'Paper side' is 'Next', 'Source Tray' is 'Current', and 'Output Tray' is '0'. Buttons for 'Edit condition', 'OK', and 'Cancel' are at the bottom.

Page Size

Available options are Executive, Letter, Legal, Ledger, Letter - no margins, Legal - no margins, Ledger - no margins, A5, A4, A3, Monarch, Com-10, DL, C5, and B5.

Note: For custom page dimension specification, user-specific specifications can be added to the eFORMzCfg_User.xml file. For further information see *Appendix A: User Configuration Files*.

Orientation

Available options are Portrait, Landscape, Reverse Portrait and Reverse Landscape.

SECTION 1

Lines Per Inch

The number of data lines to every vertical inch (default 6.0).

Form - Top and Left offsets

Controls the top and left starting position of a project form (values in decipoints).

Data - Top and Left offsets

Controls the top and left starting position of a data file (values in decipoints).

Data Viewer Size

Adjusts the Data Viewer window display to match that of the form (*Match page size*) or a customized view (*Fixed*) for data that extends beyond the form boundaries.

Exclude Data

When selected, the data output will be excluded from the form in display and printed output.

Overlay on Previous Form

Form overlay. When selected, the form will be placed on top of the previous form in the Project Tree.

Paper side

Sets the order of the forms when printed. Options: Next, Front, and Back.

Source Tray

(Default is Current Tray) Allows you to specify the tray from which the form will be pulled (i.e. Tray 1, Tray 2, Tray 3, and Tray 4). Based on the escape codes for your printer, the associated tray will need to be specified. For example, <ESC> & 1 x H, where 'x' is the value represented by the tray on the specific printer. For exact escape codes pertaining to your printer, please refer to your printer manual.

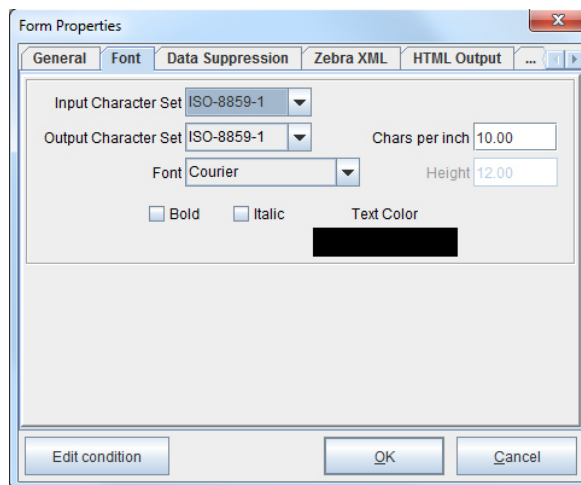
Output Tray

(Default is Current Tray) Allows you to specify the tray the form will reside in after being printed. Based on the escape codes for your printer, the associated number will need to be specified. For exact escape codes pertaining to your printer, please refer to your printer manual. <ESC> & I x G where 'x' is the value represented by the destination tray value. Note: 'l' = lower case 'L'.

Suppress output when empty

When selected, empty and/or blank pages are not printed. Default is unchecked.

Font Tab



Character Set

(Default is Roman 8) All numbers, letters, and symbols associated with a given coding system.

Font

(Default is Courier) Collection of letters, numbers, and symbols that share a distinctive appearance.

Chars per inch

Sets the number of characters per inch horizontally. Used in conjunction with fixed fonts.

SECTION 1

Height

Vertical size of a character. Used in conjunction with proportional fonts.

Bold

Changes character(s) to appear heavy with a dark typeface.

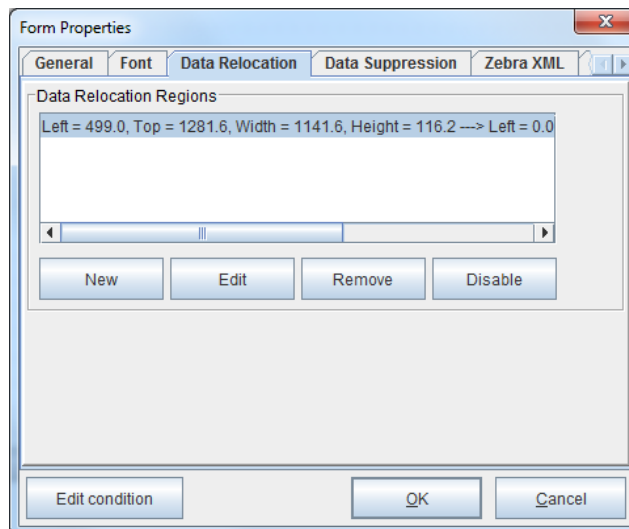
Italic

Changes character(s) to appear with a slant upward and to the right.

Text Color

Output data color when printed.

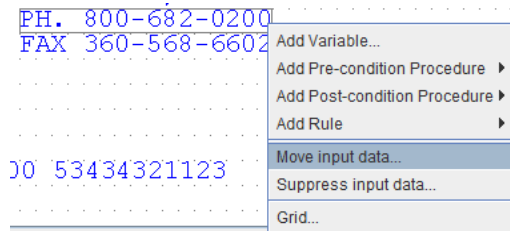
Data Relocation Tab



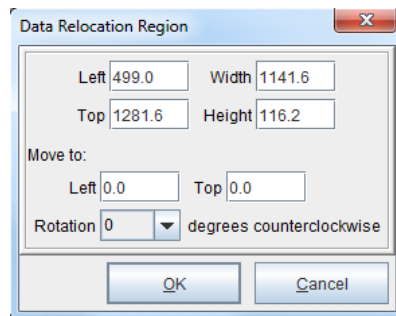
Data Relocation is the relocation of incoming data based on coordinate values. In the Input Data Viewer, the input data that is moved will be shown by a box with a green arrow. This data will be moved to appear in the specified coordinate location in the Form and Data Viewer.

Menus and Project Properties

To relocate input data, define the coordinate values from the Data Relocation tab in the Form Properties, or right click and drag the incoming data region and select *Move input data*:

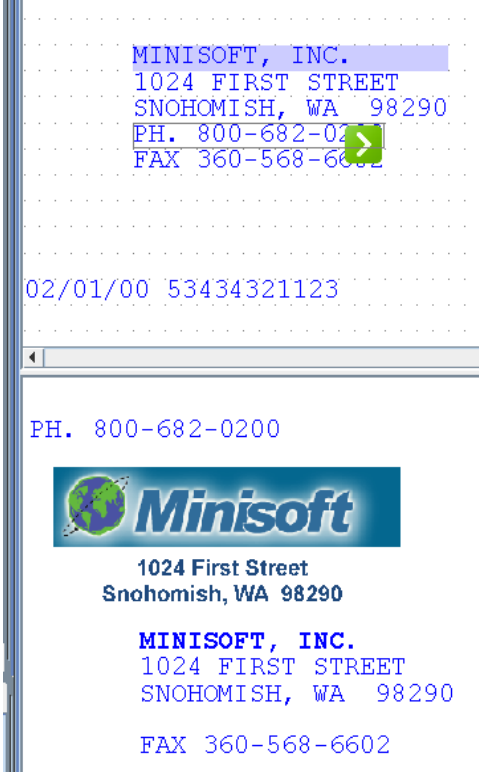


The *Data Relocation Region* dialog box will appear. Specify the coordinates, apply any rotation value and select OK.



SECTION 1

The input data that is moved will be shown by a box with a green arrow in the Input Data Viewer. This data will be moved to appear in the specified coordinate location in the Form and Data Viewer:




The screenshot shows a data entry form with a dotted grid background. The text is as follows:

MINISOFT, INC.
1024 FIRST STREET
SNOHOMISH, WA 98290
PH. 800-682-0200
FAX 360-568-6602

02/01/00 53434321123

PH. 800-682-0200



1024 First Street
Snohomish, WA 98290

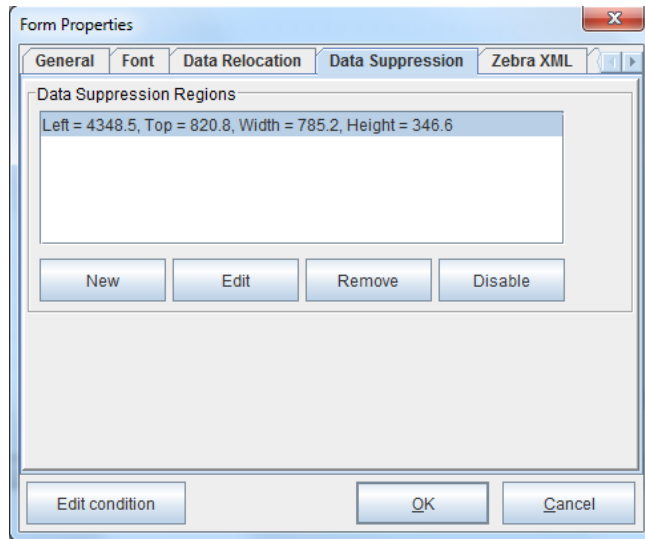
MINISOFT, INC.
1024 FIRST STREET
SNOHOMISH, WA 98290

FAX 360-568-6602

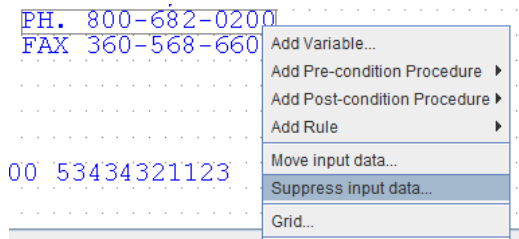
A green arrow points to the phone number '800-682-0200' in the first section of the form.

Cross-Reference: For more information on relocating data, view the following tutorial: <http://support.minisoft.com/index.php/data-relocation/>

Data Suppression Tab



Incoming data can be suppressed based on coordinate values. Data within these regions will no longer appear in your Form and Data Viewer. Data suppression regions will appear as an outline with a red 'X' in the Input Data Viewer. To add a suppression region, define the coordinate values from the Data Suppression tab in the Form Properties, or right click and drag the incoming data region and select *Suppress input data*:

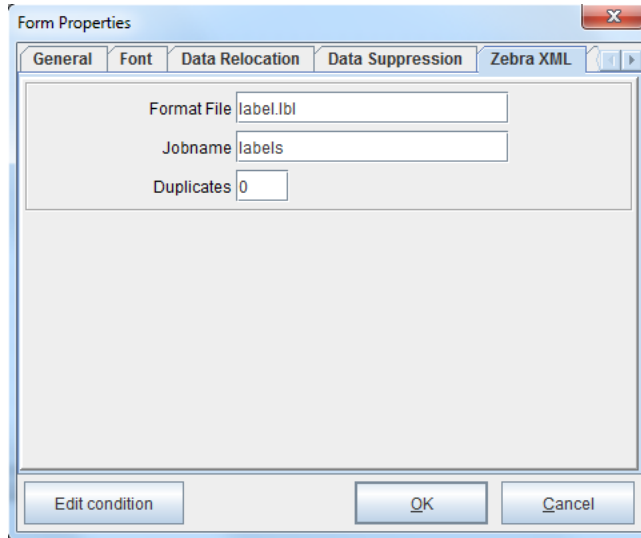


The Data Suppression Region in the Input Data Viewer:



Cross-Reference: For more information on suppressing data, view the following tutorial: <http://support.minisoft.com/index.php/data-suppression/>

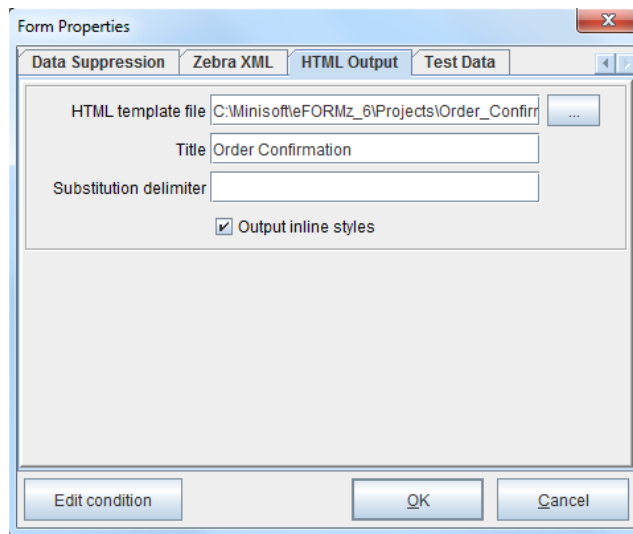
Zebra XML Tab



The Zebra XML tab allows you to generate output in Zebra XML format to send to a Zebra XML supported printer.

When the output file from eFORMz is sent to the Zebra printer it uses the *Format File* name as the reference to which template file to use within the printer (similar to macros in a laser printer). Create the label design using Zebra's XML Designer software and transfer the label design to the printer prior to printing. From the eFORMz project file create the xml tagged values that are referenced in the Zebra template file. *Jobname* is a unique name assigned to the output and *Duplicates* is the number of copies to print.

HTML Output Tab



The HTML Output tab references an external HTML file as a template for project creation. It is used exclusively with eDIRECT PLUS.

Cross-Reference: For more information on referencing an external HTML file, view the following tutorial:

<http://support.minisoft.com/index.php/referencing-external-html-files/>

SECTION 1

Test Data Tab

The screenshot shows a 'Form Properties' dialog box with the 'Test Data' tab selected. The dialog contains a list of variables and their corresponding test data values. The 'EEmailAddress' variable is set to 'test@minisoft.com'. Other variables like 'OrderNumber', 'ShipAddressLines', 'Qty', 'Description', and 'Amount' are currently empty. The dialog has 'Edit condition', 'OK', and 'Cancel' buttons at the bottom.

Variable	Test Data Value
EEmailAddress	test@minisoft.com
OrderNumber	
ShipAddressLines	
Qty	
Description	
Amount	

The Test Data tab allows the user to specify a constant value to act as sample data for a given variable.

Cross-Reference: For more information on using test data, view the following tutorial: <http://support.minisoft.com/index.php/test-data/>

SECTION 2

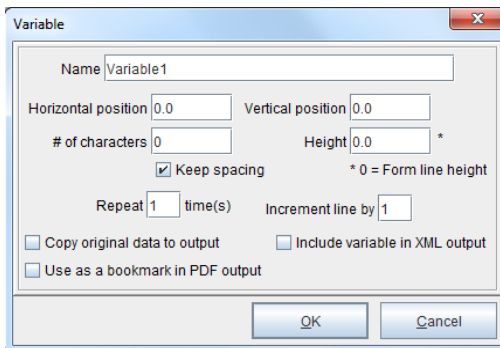
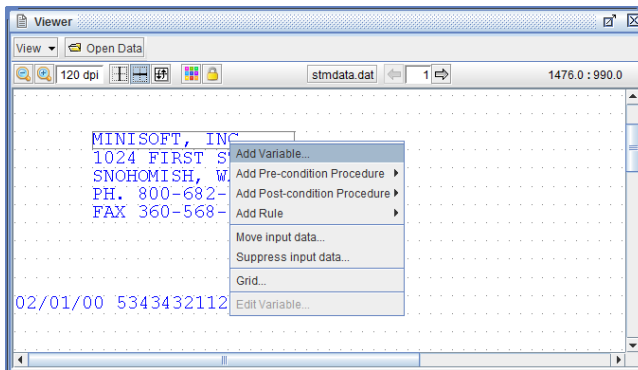
Data Customization

Variables

Variable Types

Positional Data Block Variables

From the eFORMz Input Data Viewer, select a string of characters (left click + drag), right click and select Add Variable.



Variable options include:

Name

Variable name as displayed in the project layout.

Horizontal/Vertical Position

The Variable location in decipoints (180 decipoints = 1/4 inch).

of characters

The number of characters included in a Variable.

Height

Variable height. Allows you to customize the Variable line height.

**o=Form line height*

Sets the line height to the default of the form.

Keep Spacing

When checked, eFORMz inserts spaces in the Variable value. This occurs when things on the same line are positioned by a cursor command other than spaces, and there is empty space between them. When *Keep spacing* is unchecked, no spaces are inserted in the empty space.

Repeat

The number of data lines a Variable will be implemented.

Increment line by

Enables the skipping of lines. A value of one (1) applies a Rule or Function to every line selected, a value of two (2) applies a Rule or Function to every other line, and so on.

Copy original data to output

(Default is unchecked) When checked leaves the selected characters in their original position in the eFORMz Viewer.

Include Variable in XML output

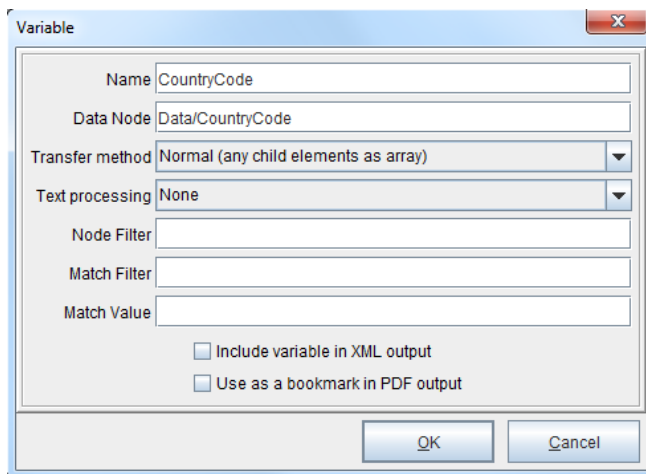
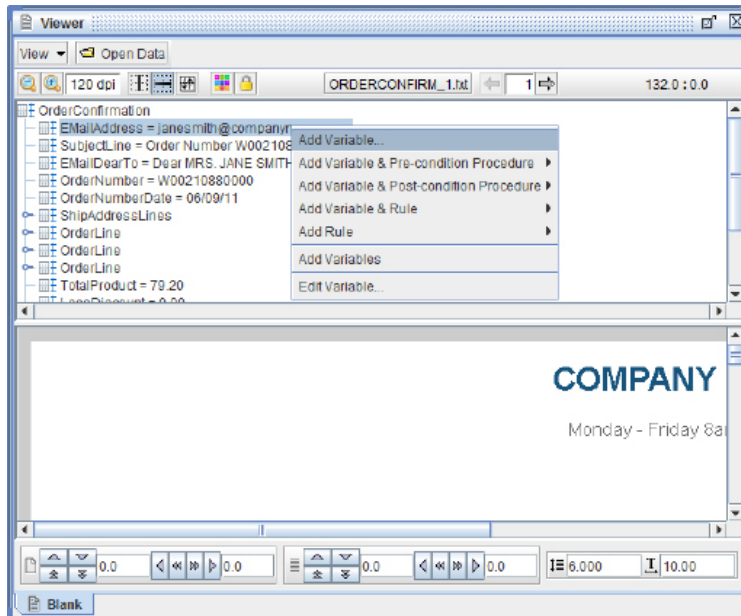
For use with eDIRECT when checked, default unchecked. Variable names cannot include special characters such as !, @, \$, &, * when this option is selected. This selection option only pertains to projects configured using the 'old' eDIRECT Wizard method of generating output.

Use as a bookmark in PDF output

Variable will be bookmarked for pdf file output.

Node Data Block Variables

For use with Extensible Markup Language (XML) data, this Form Variable is taken from the node location of the data in relation to parent or child elements in the data node tree. From the eFORMz Input Data Viewer, select the node, *right click* and select *Add Variable*.



eFORMz Composer Data Customization

Variable options include:

Name

Variable name as displayed in the project layout.

Data Node

The node location of the Variable in relation to parent or child elements in the data node tree.

Transfer method

There are seven options:

Normal (any child elements as array)

(Default) Will display any child elements as their own array, or indexed row of data within the parent element.

Count of child elements

Provides a count of the child elements contained within the parent element.

Sum of child elements' data

Calculates the sum of the child elements's data contained within the parent element.

Concatenation of child elements' data

Concatenates all of the child elements's data contained within the parent element.

Concatenation of child elements' data (comma delimited)

Concatenates all of the child elements's data contained within the parent element. Each element's value is delimited by a comma.

Concatenation of child elements' data (new line delimited)

Concatenates all of the child elements's data contained within the parent element. Each element's value is delimited by a new line.

Value of child node of a parent node

Node Filter

The Node Filter filters for an element's descendants. It allows grandchildren and great grandchildren to be parsed out.

Match Filter

Matches against the child node to retrieve the value of an XML node conditionally.

SECTION 2

Match Value

Match Value is a wildcard to match against to retrieve the value of an XML node conditionally. When a match is found, the corresponding node value from “Node Filter” is used, with the exception of “Value of child node of parent node”.

Include Variable in XML output

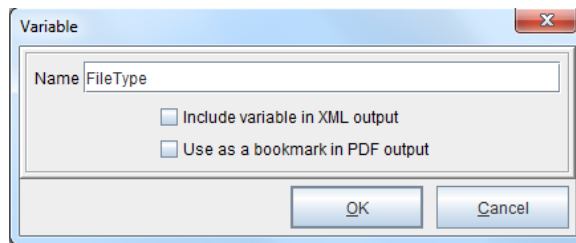
For use with eDIRECT output. The default is unchecked. If checked, Variable names cannot include special characters such as !, @, \$, *.

Use as a bookmark in PDF output

Variable will be bookmarked for PDF file output.

For activeForms Input

Used in conjunction with activeForms, this Form Variable allows the bypass of setting the “ID” property of a Document Template input field.



The image shows a dialog box titled "Variable" with a close button (X) in the top right corner. Inside the dialog, there is a text input field labeled "Name" containing the text "FileType". Below the input field, there are two checkboxes, both of which are unchecked. The first checkbox is labeled "Include variable in XML output" and the second is labeled "Use as a bookmark in PDF output". At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

Variable options include:

Include Variable in XML output

For use with eDIRECT output. The default is unchecked. If checked, Variable names cannot include special characters such as !, @, \$, *.

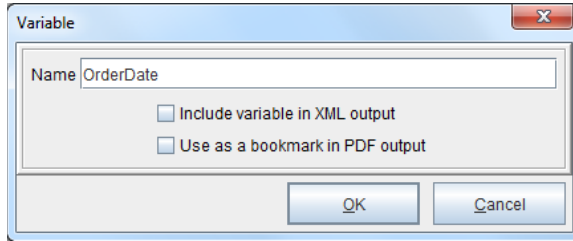
Use as a bookmark in PDF output

Variable will be bookmarked for PDF file output.

Queue Monitor Variable

This Form Variable is used to retrieve data from a specified queue. The data node name is inferred from the Variable name i.e.: "Parameters/<VariableName>"

"<root>/QMVariables/<VariableName>"



Variable options include:

Include Variable in XML output

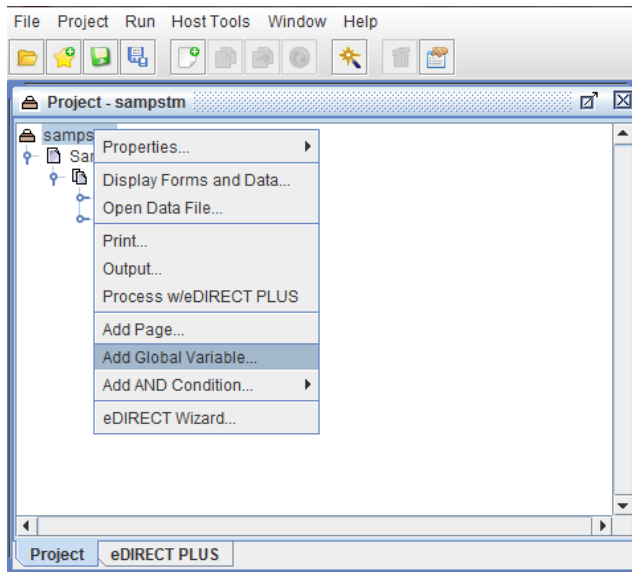
For use with eDIRECT output. The default is unchecked. If checked, Variable names cannot include special characters such as !, @, \$, *.

Use as a bookmark in PDF output

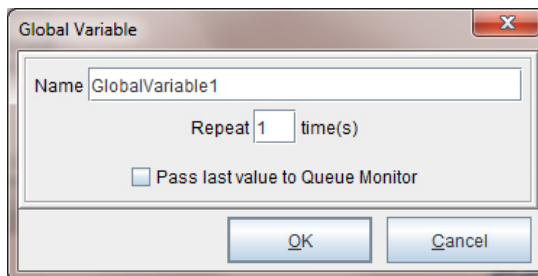
Variable will be bookmarked for PDF file output.

Global Variables

A Global Variable is a Project-level Variable. It is capable of capturing and applying text and/or numeric values from one page across multiple pages of output. To add a Global Variable to a project, right click the Project name and select *Add Global Variable*:



The Global Variable dialog box appears. Enter the global Variable name and then enter the number of lines the global Variable will need to repeat.

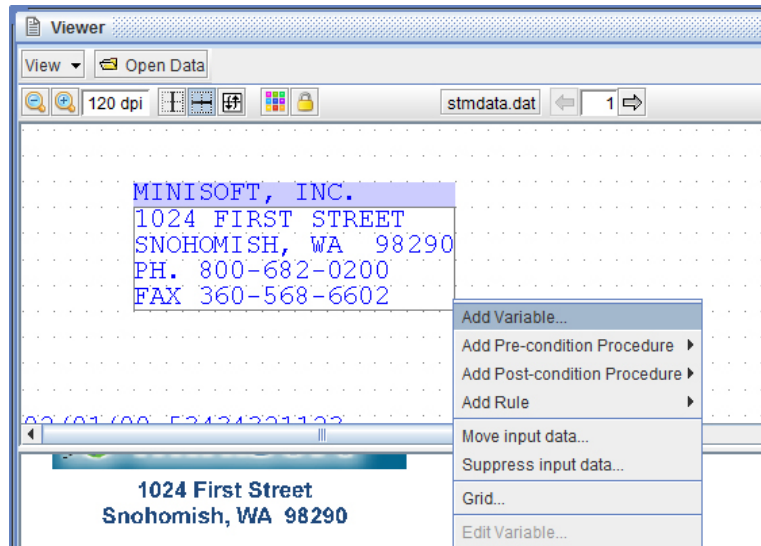


eFORMz Composer Data Customization

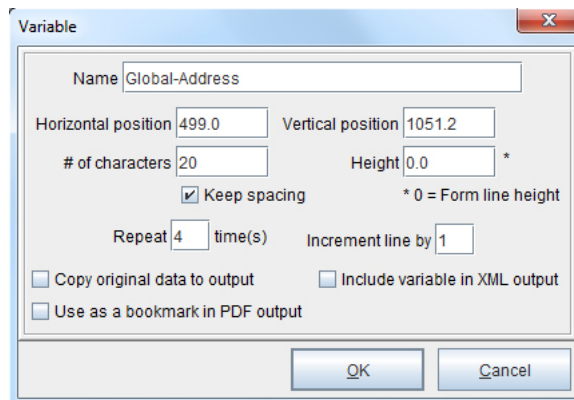
How to implement

The Global Variable can be populated with a form Variable and/or add customized Function.

1. Create a form Variable to implement as a Global Variable from the Viewer window by right-mouse clicking and selecting *Add Variable*.



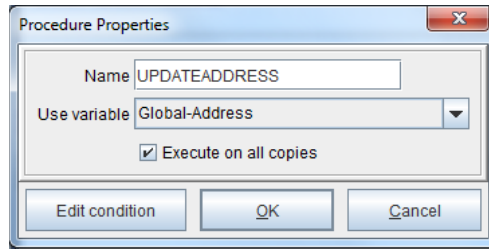
2. Enter a Variable name and select *OK*.



3. Create a Procedure that sets the value of the global Variable to your new form Variable. From the Current Project window *right click* your selected form and select *Add Pre-condition Procedure > Set Variable*.

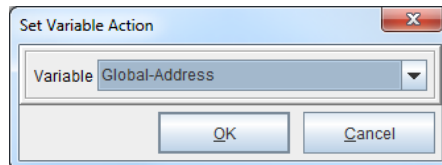
SECTION 2

4. The Procedure Properties dialog box displays. Enter a Procedure name and select the form Variable you created earlier. Select *OK*.



The screenshot shows the 'Procedure Properties' dialog box. It has a title bar with a close button (X). The main area contains a 'Name' text box with 'UPDATEADDRESS' entered, a 'Use variable' dropdown menu with 'Global-Address' selected, and a checked checkbox labeled 'Execute on all copies'. At the bottom, there are three buttons: 'Edit condition', 'OK', and 'Cancel'.

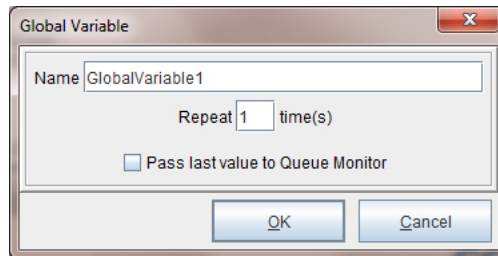
5. The Set Variable Action dialog box displays. Select the Global Variable created earlier.



The screenshot shows the 'Set Variable Action' dialog box. It has a title bar with a close button (X). The main area contains a 'Variable' dropdown menu with 'Global-Address' selected. At the bottom, there are two buttons: 'OK' and 'Cancel'.

The Procedure now sets the global Variable. A *condition* on the Procedure must also be set in order for the Procedure to implement across multiple pages. The condition must match only on the first page.

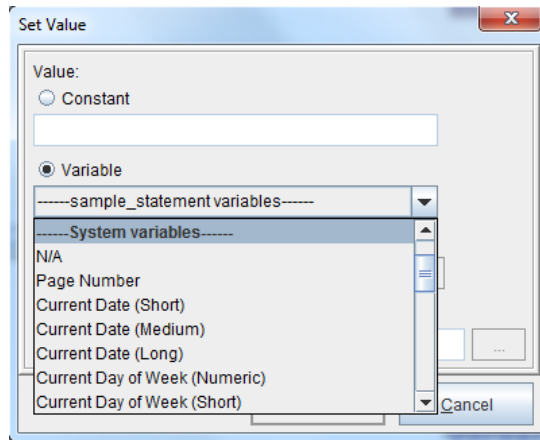
6. If you would like to implement an action based on the global Variable, proceed with creating an action using the global Variable on the selected form.



The screenshot shows the 'Global Variable' dialog box. It has a title bar with a close button (X). The main area contains a 'Name' text box with 'GlobalVariable1' entered, a 'Repeat' text box with '1' entered and 'time(s)' to its right, and an unchecked checkbox labeled 'Pass last value to Queue Monitor'. At the bottom, there are two buttons: 'OK' and 'Cancel'.

System Variables

System Variables are dynamic and implemented at runtime. System Variables can be added using Procedures and Functions. After a Variable has been created use a Set Value to implement a System Variable:



Available system Variables:

System Variables	
N/A	Inserts an empty string
Page Number	Example: xx
Current Date (short)	Example: mm/dd/yy
Current Date (Medium)	Example: mmm dd, yyyy
Current Date (Long)	Example: m...m dd, yyyy
Current Day of Week (Numeric)	Sunday = 1, Tuesday = 3
Current Day of Week (Short)	Example: Mon, Tue, Wed, Thu, Fri, Sat, Sun
Current Day of Week (Long)	Example: Monday, Tuesday, Wednesday, etc...
Current Day of Month	Example: dd
Current Month (Numeric)	Example: mm
Current Month (Short)	Example: mmm
Current Month (Long)	Example: m...m
Current Year	Example: yyyy

SECTION 2

Current Time (Short)	Example: hh:mm am/pm
Current Time (Medium)	Example: hh:mm:ss am/pm
Current Time (Long)	Example: hh:mm:ss am/pm timezone
Current Hour (24)	Example: hh
Current Hour (12)	Example: hh
Current Minute	Example: mm
Current Second	Example: ss
Current Millisecond	Example: mm
AM/PM	Example AM or PM
TimeZone (Short)	Example: PDT
TimeZone (Long)	Example: Pacific Daylight Time
TimeZone (RFC 822)	GMT representation
Current Time (Timestamp)	Returns the current time as the number of milliseconds since 1/1/1970
Data File Name	Incoming data file name, for example: stmdata.dat
Full Data File Name	Incoming data file name including data file path, for example: path...\stmdata.dat
Data File Timestamp	Returns the data files last modified time as the number of milliseconds since 1/1/1970
true	Sets the value to "true".
Raw Page Data	Contains all the data from the input page concatenated together. The Variable contents are not calculated unless they are actually used

Variable Functions

eFORMz supports the following standard Variable Functions:

- ◆ Concatenate
- ◆ Insert String
- ◆ Set Value
- ◆ Substring
- ◆ Replace
- ◆ Choose one of
- ◆ Set Value from a Declared Constant
- ◆ Set Value from a Keyed Declared Constant
- ◆ Translate
- ◆ Parse Date
- ◆ Right Trim
- ◆ Left Trim
- ◆ Uppercase
- ◆ Lowercase
- ◆ URL Encoder
- ◆ URL Decoder
- ◆ UPC-EAN Check Digit
- ◆ Table Lookup
- ◆ SQL Lookup
- ◆ Does File Exist

SECTION 2

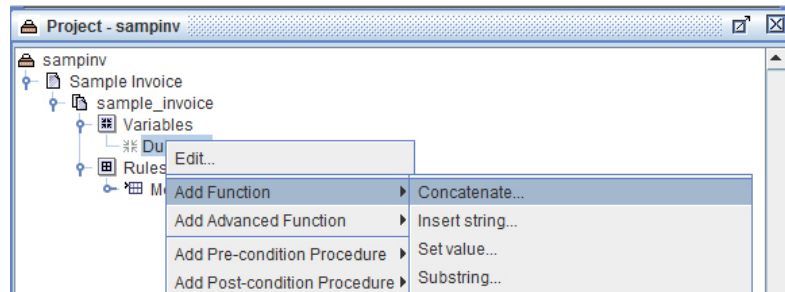
eFORMz supports the following advanced Variable Functions:

- ◆ eVS Barcode Data Generator
- ◆ SmartLabel Barcode Data Generator
- ◆ Bulk Concatenate
- ◆ Base64 Decode
- ◆ Base64 Encode
- ◆ Regex Replace
- ◆ Modulo
- ◆ HashSHA1
- ◆ String Length
- ◆ String Position

Standard Functions

Concatenate

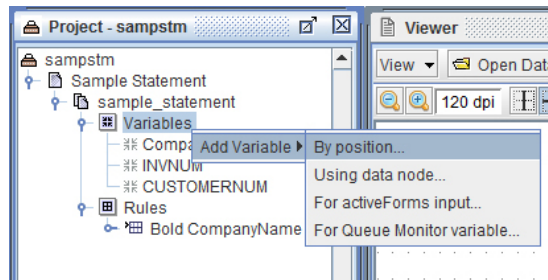
Concatenate allows the user to combine two or more character strings to make one:



How to implement

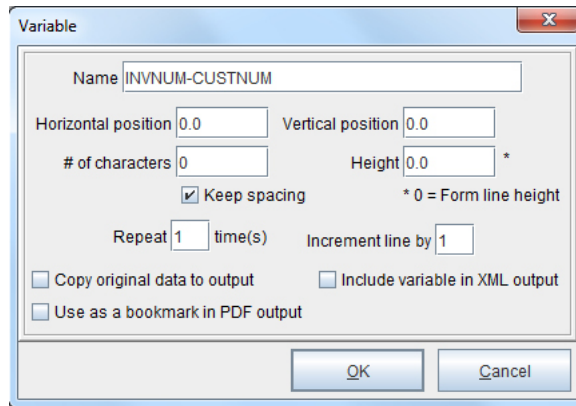
Concatenate will allow you to combine two or more character strings to make one. To implement a concatenate Function:

1. From the eFORMz Input Data Viewer, create a Variable for each string that will make up the concatenated string.
2. A third Variable can be created to combine the two Variables together if you wish to use each of the first two separately and for other Rule options. From the Variables placeholder, right click, select *By Position*:

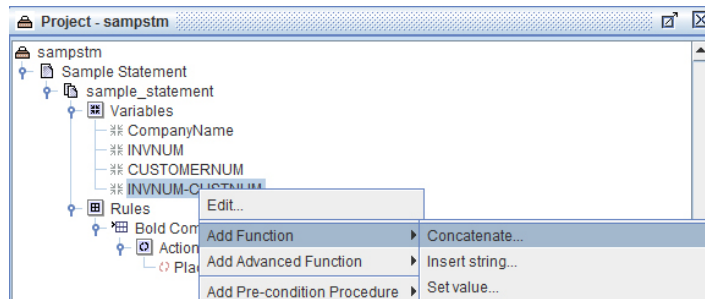


SECTION 2

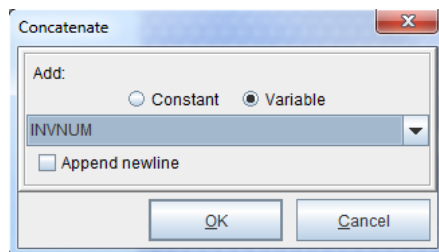
3. Enter a new Variable name and then select *OK*.



4. From the Current Project window right click your newly created Variable and select *Add Function > Concatenate*:



5. From the Concatenate dialog box select the Variable option and then select the first Variable you would like to join together.

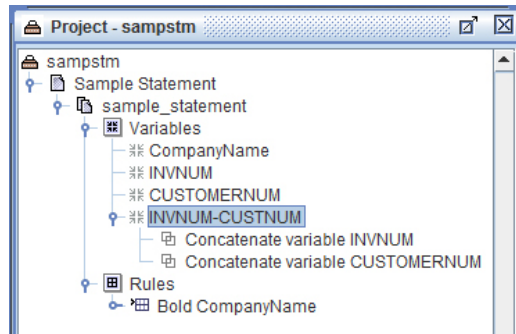


Select *OK*.

NOTE: The *Append newline* feature will only work in conjunction with the *Wordwrap* option of the *Place Text* action.

eFORMz Composer Data Customization

6. Follow the same Procedure for concatenating the second Variable.
7. When complete your newly created Variable will display both concatenated Variables.



TIP: To create a continuous string that combines multiple variables, create a single variable that contains multiple concatenate functions. By default, a variable containing multiple concatenates functions as one continuous string. To allow for line breaks between the concatenate functions select the *Append newline* option. The *Append newline* option works in conjunction with the wordwrap option of the Place Text Rule.

8. Next, create a new Rule to display your newly configured string. From the eFORMz Form Viewer, *right click* and select *Add Rule > 1D Barcode (or another Rule type)*.
9. The *Rule Properties* dialog box will display. Enter a new Rule name and click *OK*.

SECTION 2

10. The *Barcode Action* dialog box will display. Select the a barcode type and enter the desired width and height. Once complete, click *OK*.

Barcode Action

Position

Barcode type: 3OF9

Generate checksum

Trim data

Optimize barcode length

Barcode Color: [Black]

Horizontal position: 4963.1 Width: 10.0000

Vertical position: 553.9 Height: 300.0000

Alignment

Rotation: 0

None

Right

Center Width: 0.0

Repeat Interval Spacing*: 0.0

*0=Use Form's lines per inch

Conditional vertical increment

OK Cancel

11. A barcode will display in the eFORMz Form Viewer. The new barcode is the concatenated string created earlier with the Variable containing the multiple Functions.

Cross-Reference: For more information on Concatenate, view the following tutorial: <http://support.minisoft.com/index.php/concatenate-video-only/>

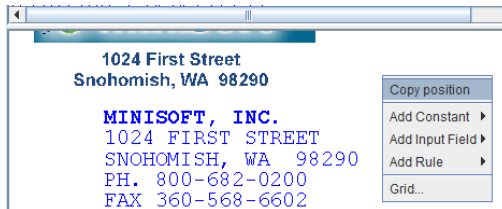
Insert String

Insert a String will insert a certain number of characters into a Variable in one of two ways:

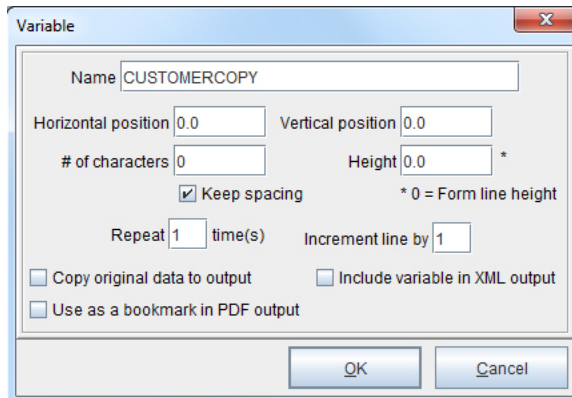
- ◆ Place and insert a string of characters in a specified position.
- ◆ Insert a string of characters attached to an existing string of data from the incoming data file.

How to implement

1. In the eFORMz Form Viewer, locate a specified position for the data string to be placed, *right click* and select *Copy Position*.

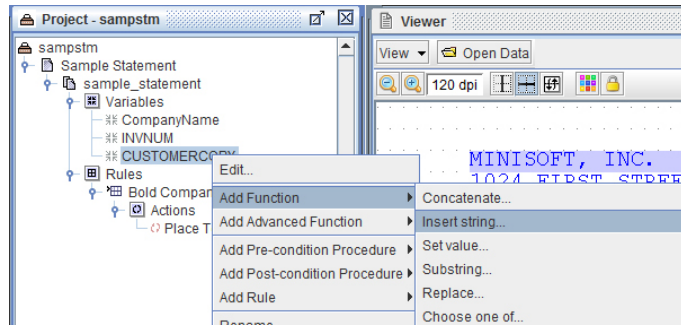


2. From the Project window *right click* the form and select *Add Variable > By Position*.
3. The Variable dialog box will display. Enter a new Variable name and click *OK*.

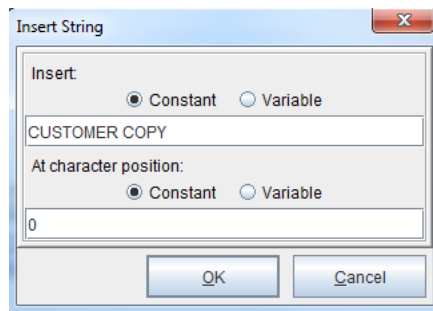


SECTION 2

- The new Variable will appear in the Current Project window. Using your mouse, right click the new Variable name and select *Add Function > Insert string*.



- The *Insert String* dialog box will display. Enter a *Constant* value, for example, CUSTOMER COPY in the *Insert > Constant* field. In the *At character position* field enter zero (0).



Click *OK*.

- Select the newly created Rule from the Current Project window and then select *Add Rule > PlaceText*.
- The *Rule Properties* dialog box will display. Enter a new Rule name and select the previously created Variable for the *Use Variable* field. Once complete, click *OK*.

eFORMz Composer Data Customization

- From the *Place Text* dialog box select *Position > Paste*. The position previously selected will be pasted into the *Horizontal* and *Vertical* position fields.
Select *Change Font* if you would like any font changes to be applied.

Place Text Action

Position

Copy

Paste

Horizontal position 2135.3

Vertical position 365.5

Text Color

Underline

Style None

Thickness 7.2 Offset 7.2

Horizontal Alignment Vertical Alignment

Rotation 0

None Right Left Center

Width: 0.0

Wordwrap

Enable wordwrap

Line Width 0.0 Line Spacing* 0.0

Suppress blank lines

Repeat Interval Spacing* 0.0

*0=Use Form's lines per inch

Conditional vertical increment

Change font OK Cancel

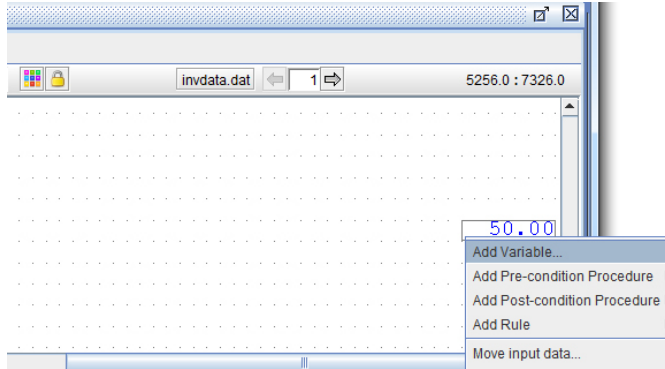
Click *OK*.

- The text string will appear in the eFORMz Viewer, Form and Data Viewer.

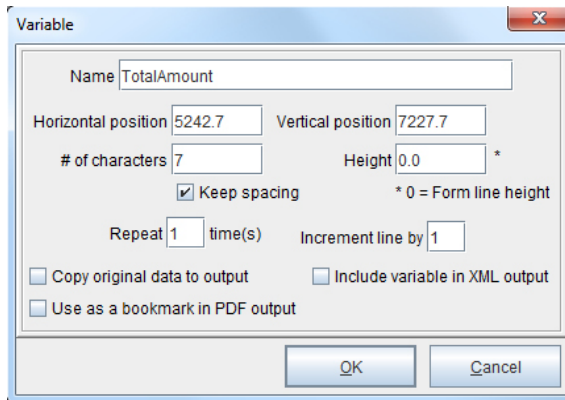
Cross-Reference: For more information on Insert String, view the following tutorial: <http://support.minisoft.com/index.php/insert-string/>

Attach String to Existing Data

1. From the eFORMz Viewer, Input Data Viewer, select the data you would like to add a string to, *right click*, and select *Add Variable*.



2. The *Variable* dialog box will display. Enter a new Variable name and click *OK*.

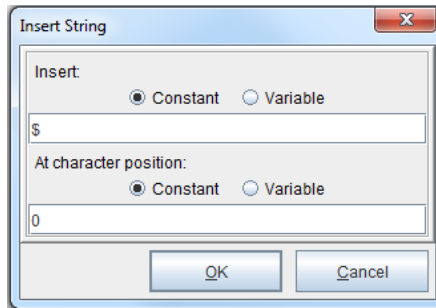


3. The newly created Variable will display in the Project window. *Right click* the Variable and select *Add Function > Insert String*.
4. The *Insert String* dialog box will display. Enter the character string you would like to attach to the selected Variable in the *Insert > Constant* field. Enter the starting character position in the *At character position > Constant* field. Examples:

Character String	Character Position	Output
\$	0	\$999,999.00
/	2	
/	5	03/15/05
.00	7	999,999.00

eFORMz Composer Data Customization

Note: Character positioning starts at zero (0).

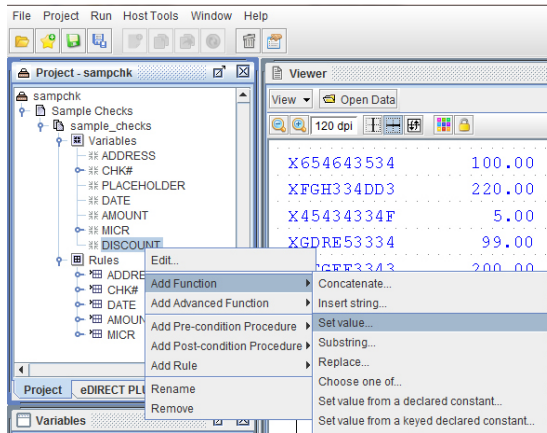


Once complete, click *OK*.

5. From the eFORMz Viewer, Form and Data Viewer, *right click* in the position you would like your new string to appear and select *Add Rule > Place Text*.
6. From the *Rule Properties* dialog box enter a new Rule name and select the previously created Variable. Once complete, click *OK*.
7. The *Place Text Action* dialog box will display. Click *OK* if you are complete with the Place Text Action dialog box.
8. Your new Rule will appear in the Current Project window and in the eFORMz Viewer, Form and Data Viewer. If the newly created string needs adjusting in the Form and Data Viewer, using your mouse, drag-and-drop the new string into position.

Set Value

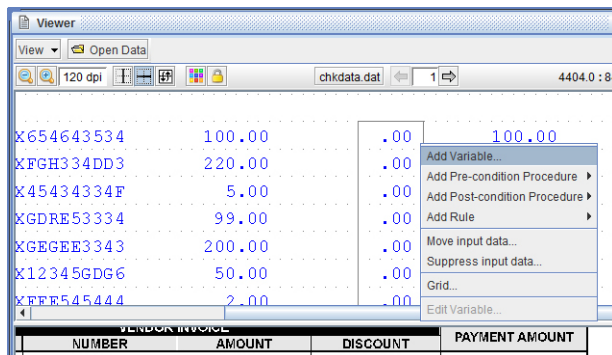
A Variable can be set to a particular value (*Constant, Variable, Image from Project & Image from File*). The newly set value will override any preceding Function or value being picked up by the Variable.



How to implement

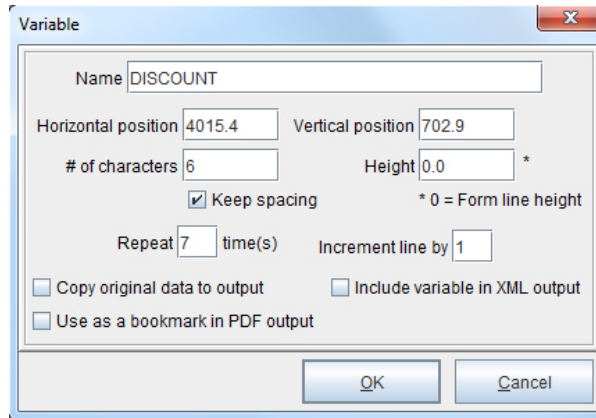
To change the value of a set of characters:

1. From the eFORMz Viewer, Input Data Viewer, select the characters you would like to change. *Right click* and select *Add Variable*.



2. The *Variable* dialog box will display. Enter a new Variable name and click *OK*.

eFORMz Composer Data Customization

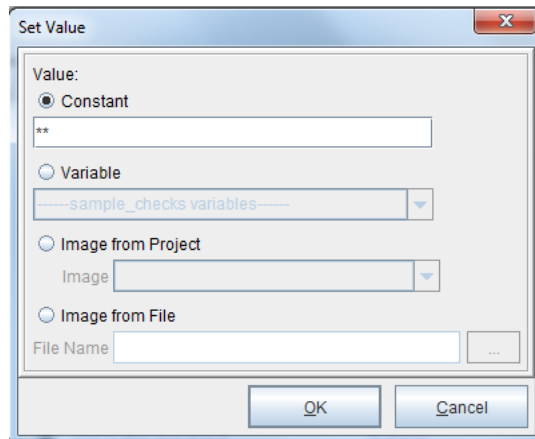


The Variable dialog box is used to define a new variable. It contains the following fields and options:

- Name: DISCOUNT
- Horizontal position: 4015.4
- Vertical position: 702.9
- # of characters: 6
- Height: 0.0 * (where * = Form line height)
- Keep spacing
- Repeat: 7 time(s)
- Increment line by: 1
- Copy original data to output
- Include variable in XML output
- Use as a bookmark in PDF output

Buttons: OK, Cancel

3. The newly created Variable will display in the Current Project window. Using your mouse, *right click* the new Variable from the Current Project window and select *Add Function > Set Value*.
4. The *Set Value* dialog box displays. For the *Value type* select *Constant* and enter a value you would like to set the characters to in the constant value field.



The Set Value dialog box is used to define the value for a variable. It contains the following options and fields:

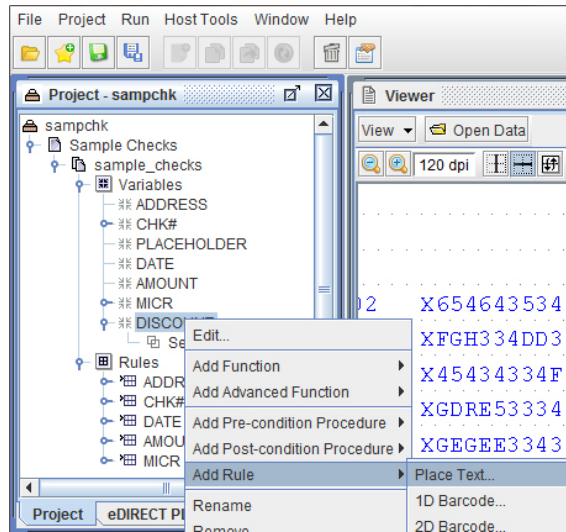
- Value:
 - Constant: **
 - Variable: sample_checks variables
 - Image from Project: Image
 - Image from File: File Name

Buttons: OK, Cancel

Once complete, click *OK*.

SECTION 2

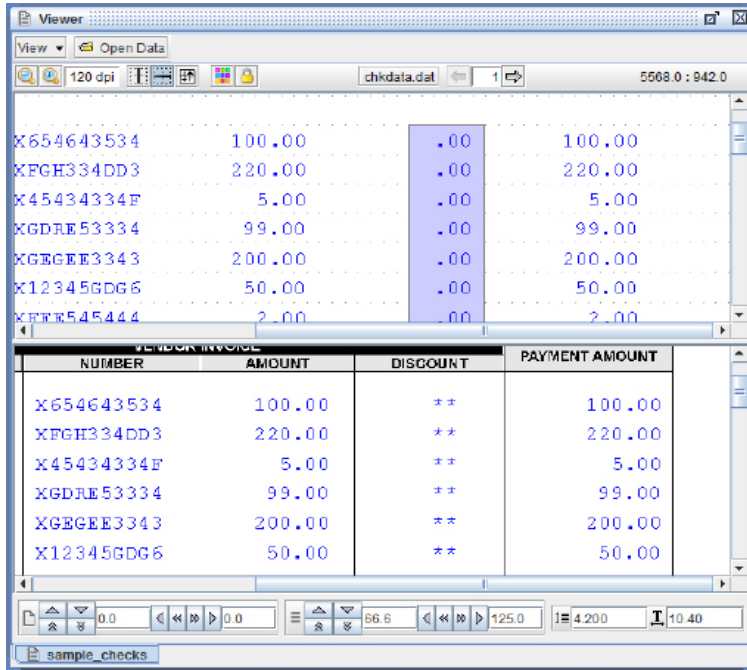
- The Set Value Function will display in the Current Project window. From the Current Project window select the new Variable, *right click*, and select *Add Rule > Place Text*.



- The *Rule Properties* dialog box will display, enter a new Rule name and click *OK*.
- The *Place Text Action* dialog box will display. Without making any changes click *OK*.

eFORMz Composer Data Customization

- The eFORMz Form Viewer displays the character change while the Input Data Viewer displays the original data format.

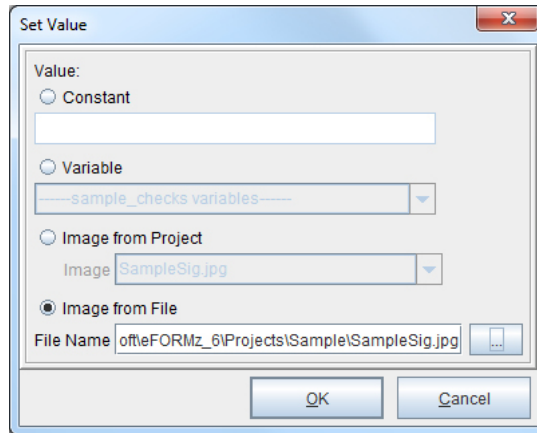


Cross-Reference: For more information on Set Value, view the following tutorial: <http://support.minisoft.com/index.php/set-value/>

Setting an Image Value

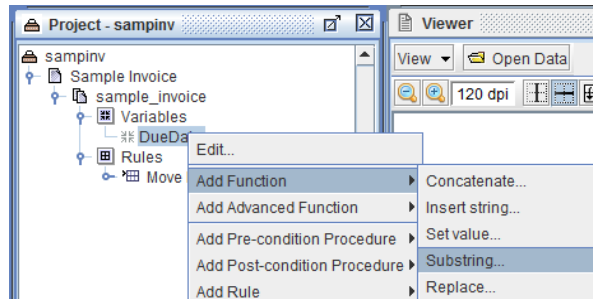
The Set Value Function can also be set to an image. Apply an image in one of the following ways:

- ◆ *Image from project* loads a previously saved image that was imported into the project using the *Project Properties > Images* option.
- ◆ *Image from file* loads an image from a specified folder. The image will not be stored in the project file.



Substring

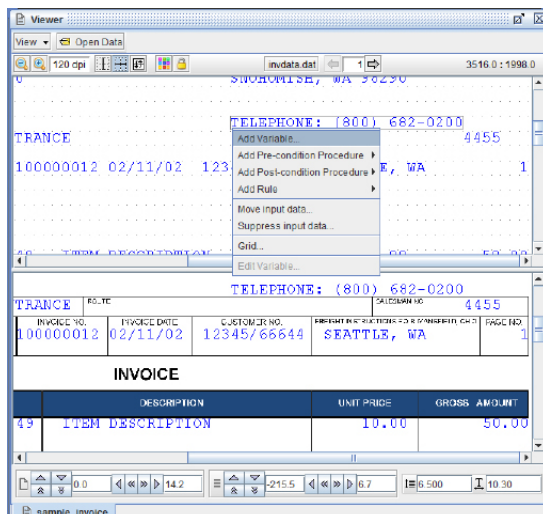
Substring extracts a string of characters from an incoming data file or a custom-built Variable:



How to implement

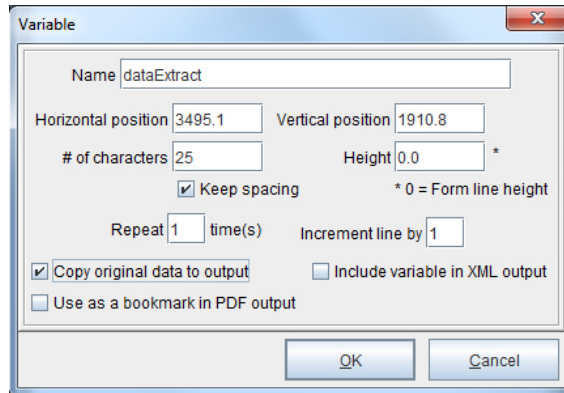
Extract a string of characters using the substring Function from your incoming data file or a custom built Variable. For example, to extract a string of characters from your incoming data file:

1. From the eFORMz Viewer, Input Data Viewer, select the data you would like to extract. Once selected, *right click* your mouse and select *Add Variable*.



SECTION 2

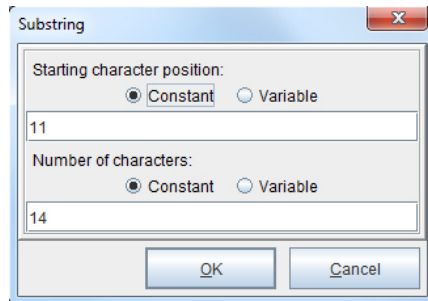
- From the *Variable* dialog box enter a new Variable name. If you would like the original data to appear in your output select *Copy original data to output*. Click *OK*.



Variable dialog box showing configuration for a new variable named "dataExtract".

- Name: dataExtract
- Horizontal position: 3495.1
- Vertical position: 1910.8
- # of characters: 25
- Height: 0.0 *
- Keep spacing
- Repeat: 1 time(s)
- Increment line by: 1
- Copy original data to output
- Include variable in XML output
- Use as a bookmark in PDF output

- Right click the newly created Variable and select *Add Function > Substring*.
- From the *Substring* dialog box enter the starting position of the string you would like to extract. Note, for this example we are extracting the entire data. Set the number of character to be extracted.



Substring dialog box showing configuration for extracting a substring.

- Starting character position: Constant (selected), Variable (unselected), value: 11
- Number of characters: Constant (selected), Variable (unselected), value: 14

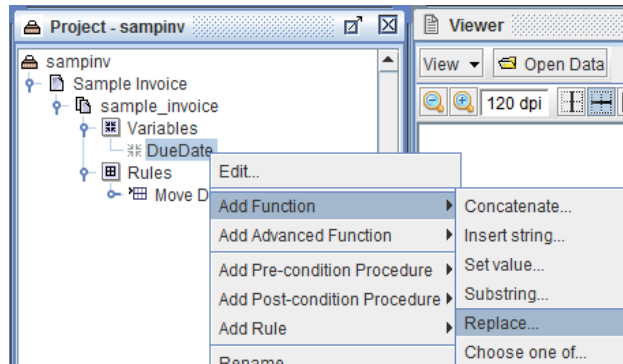
TIP: Specifying a negative value (-1) in the *Starting Character Position* field indicates to start from the end of the string. '0' is the starting position of the string.

Once complete, click *OK*.

Cross-Reference: For more information on Substring, view the following tutorial: <http://support.minisoft.com/index.php/substring/>

Replace

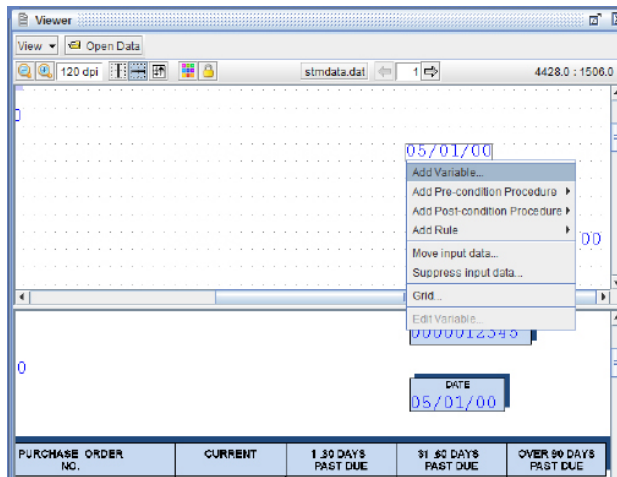
Replace a character or string of characters with another:



How to implement

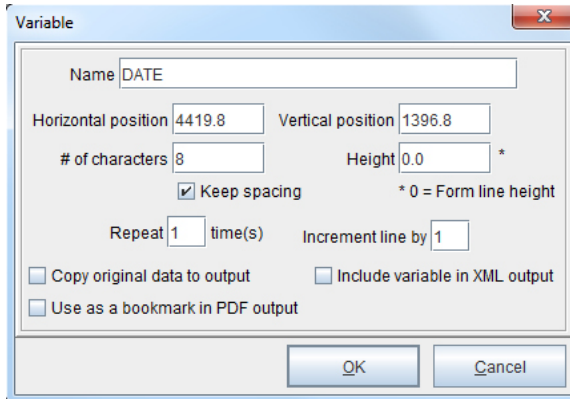
To replace a character or string of characters with another:

1. From the eFORMz Data Viewer, select the block of characters you would like to do a search and replace on. Once the characters have been selected, *right click* using your mouse and select *Add Variable*.



SECTION 2

2. The *Variable* dialog box will display. Enter a new Variable name and click *OK*.

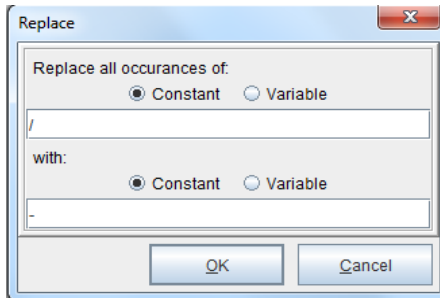


The Variable dialog box is shown with the following settings:

- Name: DATE
- Horizontal position: 4419.8
- Vertical position: 1396.8
- # of characters: 8
- Height: 0.0 *
- Keep spacing
- Repeat: 1 time(s)
- Increment line by: 1
- Copy original data to output
- Include variable in XML output
- Use as a bookmark in PDF output

3. From the Project window select the newly created Variable, *right click*, and select *Add Function > Replace*.
4. The *Replace Text* dialog box will display. From the *Replace all occurrences of > Constant* field enter the character(s) you would like to replace.

From the *with > Constant* field enter the character you would like the above character(s) to be replaced with.



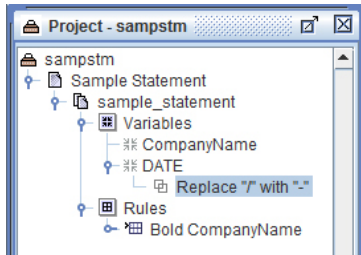
The Replace dialog box is shown with the following settings:

- Replace all occurrences of:
 - Constant
 - Variable
 - Field: /
- with:
 - Constant
 - Variable
 - Field: -

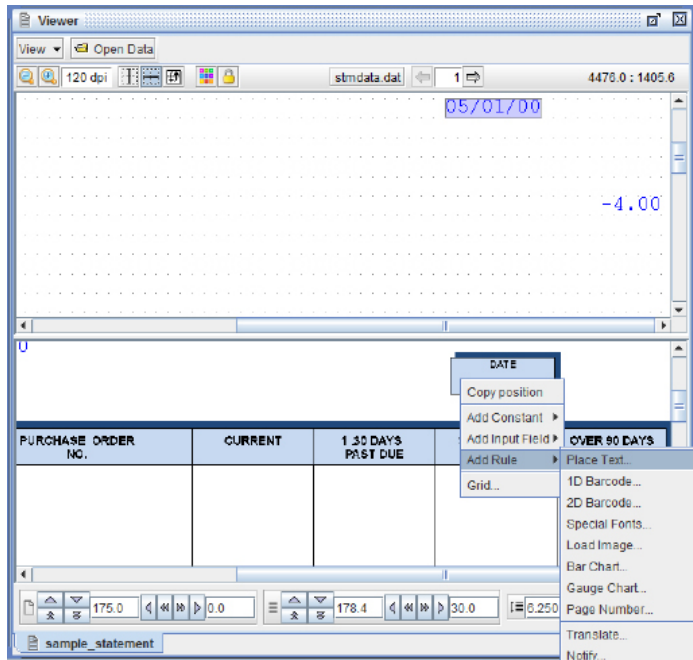
Click *OK*.

eFORMz Composer Data Customization

5. The *Replace* Function will display in the Current Project window:

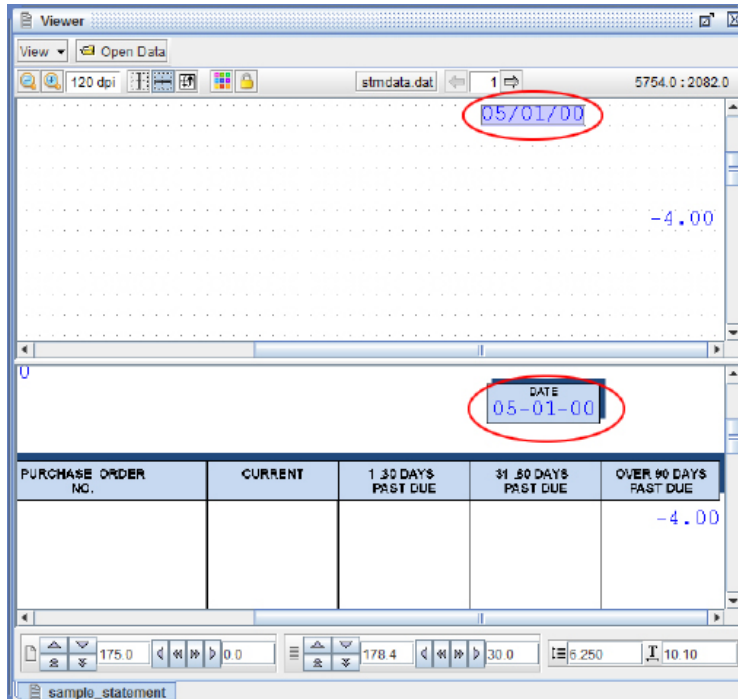


6. From the eFORMz Viewer, Form and Data Viewer, *right click* in the area you would like your new string of data to appear. Select *Add Rule > Place Text*.



SECTION 2

7. The *Rule Properties* dialog box will display. Enter a new Rule name and select the newly created Variable from the *Use Variable* field. Once complete, click *OK*.
8. The *PlaceText Action* dialog box displays. Make any necessary changes and click *OK*.
9. The new Rule action will display in the Current Project window and in the eFORMz Viewer, Form and Data Viewer.



Cross-Reference: For more information on Replace, view the following tutorial: <http://support.minisoft.com/index.php/replace-function-video-only/>

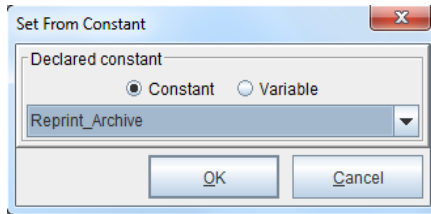
Choose One Of

The Choose One Of Function can be used to choose a specific occurrence from an occurring Variable by index or by matching against a parallel occurring Variable.

The screenshot shows a dialog box titled "Choose One Of" with a close button (X) in the top right corner. The dialog is divided into two main sections. The first section, "Choose using index", is selected with a radio button. It contains a dropdown menu labeled "Choices in variable" with the text "sample_invoice variables" and a downward arrow. Below this is a sub-section labeled "Index" with two radio buttons: "Constant" (selected) and "Variable". A text input field is positioned below these options. The second section, "Choose using match", is unselected. It contains a dropdown menu labeled "When variable" with the text "sample_invoice variables" and a downward arrow. Below this is a sub-section labeled "matches" with two radio buttons: "Constant" (selected) and "Variable". A text input field is positioned below these options. At the bottom of the dialog are two buttons: "OK" and "Cancel".

Set Value from a Declared Constant

A Declared Constant from an eFORMzCfg_User.xml file provides the value for this Variable Function.



The form of a declaration is:

```
<Constant Name="OrderNbr">
  <![CDATA[Order Number]]>
</Constant>
```

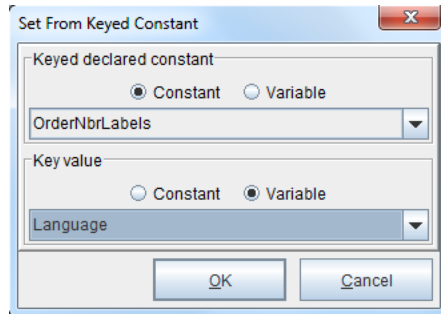
If this Function is used in a number of Projects to refer to the same Declared Constant, when the Declared Constant must change, only the eFORMzCfg_User.xml file needs to be modified, not each Project.

Note: Changes to the eFORMzCfg.xml will get overwritten as this file is updated or overwritten on upgrades and installs.

Cross-Reference: For a tutorial on using the *Set Value from a Declared Constant* Function, see the Language Substitution part of the *Document Template Mini-Manual*.

Set Value from a Keyed Declared Constant

A Keyed Declared Constant from an eFORMzCfg_User.xml file provides the value for this Variable Function.



The form of a declaration is:

```
<KeyedConstant Name="OrderNbrLabels">
  <Entry Key="EN">
    <![CDATA[Order Number]]>
  </Entry>
  <Entry Key="FR">
    <![CDATA[French Order Number]]>
  </Entry>
</KeyedConstant>
```

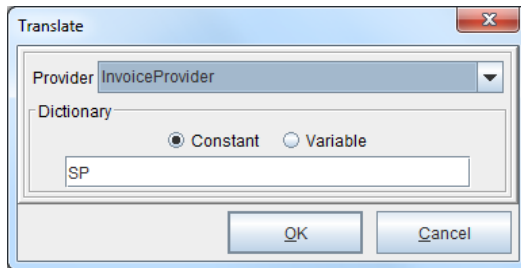
If this Function is used in a number of Projects to refer to the same Keyed Declared Constant, when the Keyed Declared Constant must change, only the eFORMzCfg_User.xml file needs to be modified, not each Project.

Note: Changes to the eFORMzCfg.xml will get overwritten as this file is updated or overwritten on upgrades and installs.

Cross-Reference: For a tutorial on using the *Set Value from a Keyed Declared Constant* Function, see the Language Substitution part of the *Document Template Mini-Manual*.

Translate

The Translate Function uses a dictionary of values set in a declared constant (Translation Provider) from an eFORMzCfg_User.xml file to automatically update a value based upon a conditional flag in a project.



The form of a declaration is:

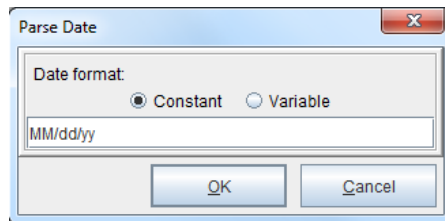
```
<Translator Name="InvoiceProvider" Class="com.minisoft.efzUtils.XMLFileTranslationProvider"  
  FileName="C:\Minisoft\efORMz_6\Projects\Dictionaries.xml"/>
```

Cross-Reference: For more information on using the Translate Function, view the following tutorial:

<http://support.minisoft.com/index.php/using-the-eformz-translation-file-editor-video-only/>

Parse Date

The Parse Date Function parses text representing a date/time into a timestamp (number of ms since 1/1/1970). The "Date format" field specifies the format of the text. It is described in detail here: docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html

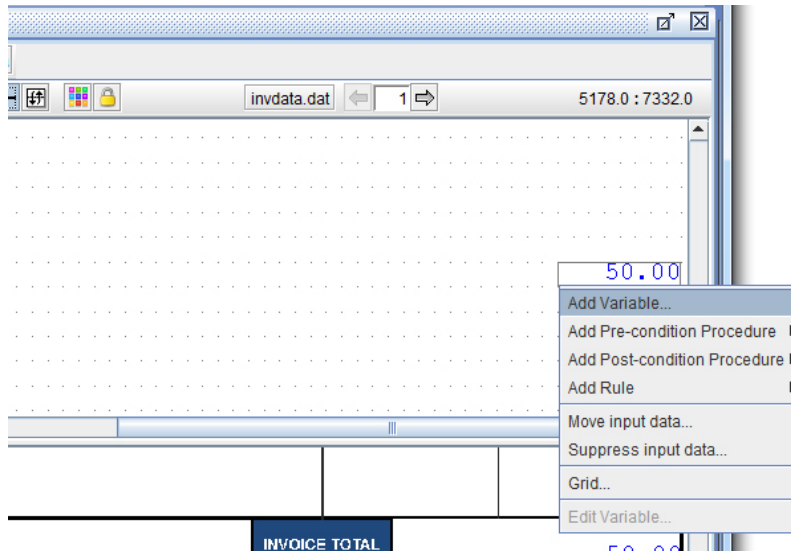


Cross-Reference: For an instance of this Function being used, view the following tutorial:

<http://support.minisoft.com/index.php/date-calculation-procedure-video-only/>

Right and Left Trim

Trim extra blank spaces from the right or left of a Variable using the Right and Left Trim Functions.



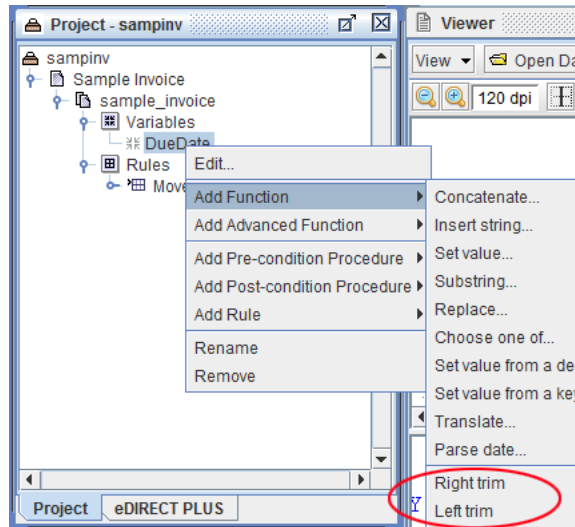
How to implement

Trim extra blank spaces from the right or left of a Variable using the Right and Left Trim Functions. To implement a trim:

1. From the eFORMz Data Viewer, select the data you would like to apply a trim format to.
2. Once the string of characters has been selected, using your mouse, *right click* and select *Add Variable*.
3. The *Variable* dialog box will display. Enter a new Variable name and click *OK*.

eFORMz Composer Data Customization

- The new Variable will appear in the Project window. Using your mouse, *right click* the Variable and select *Add Function > Left Trim* (or *Right Trim*).



- Select the newly created Variable from the Project window. *Right click* and select *Add Rule > Place Text*.
- The *Rule Properties* dialog box displays. Enter a new Rule name and click *OK*.
- Without changing the horizontal and vertical positions, click *OK*.
- The string adjustment will display in the eFORMz Viewer Form Viewer.

Cross-Reference: For more information on Right and Left Trim, view the following tutorial: <http://support.minisoft.com/index.php/right-left-trim-video-only/>

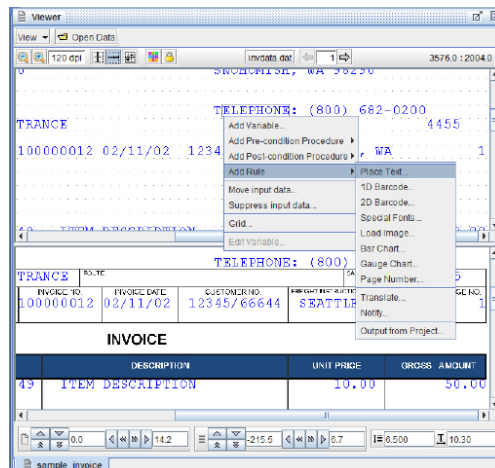
Lowercase and Uppercase

The Lowercase Function makes all text characters in a Variable lowercase, while the Uppercase Function makes all text characters in a Variable uppercase.

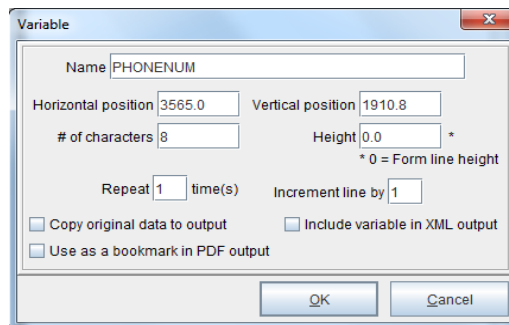
How to implement

To implement a lowercase or uppercase Function to a Variable:

1. From the eFORMz Data Viewer, select the range of data you would like to transform. Using your mouse *right click* and select *Add Rule > Place Text*.

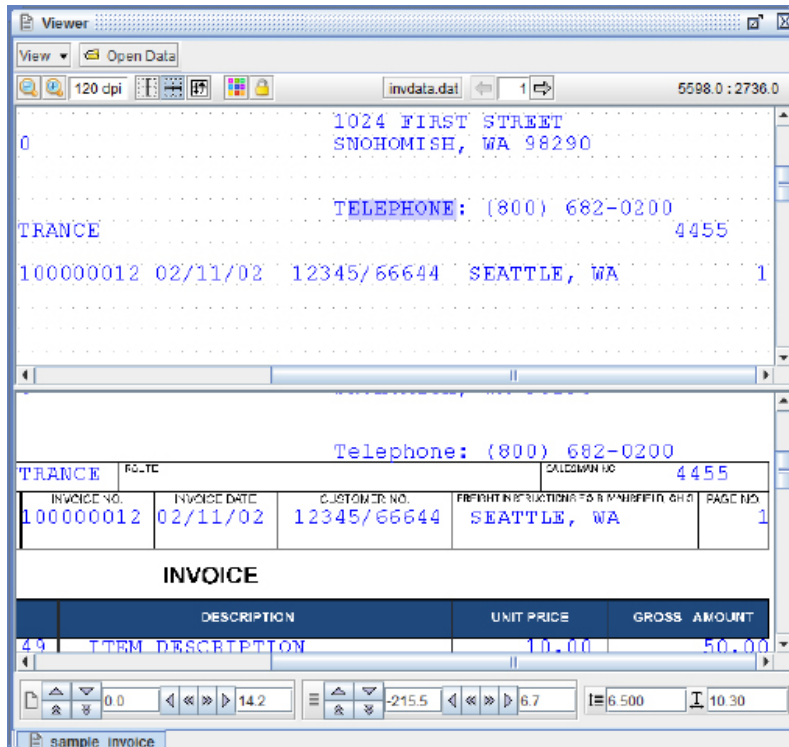


2. The *Variable* dialog box will display. Enter a new Variable name and click *OK*.



eFORMz Composer Data Customization

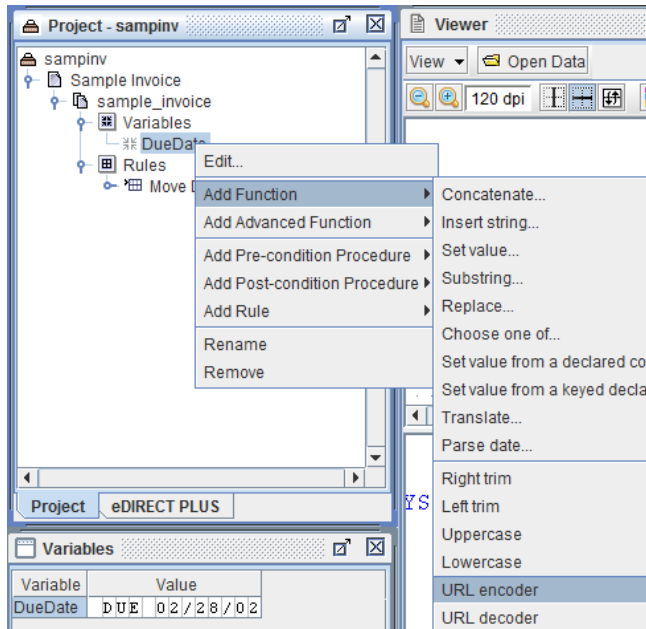
3. From the Rule Properties dialog box enter a new Rule name and click *OK*.
4. From the *Place Text Action* dialog box, keep the existing Horizontal and Vertical position and click *OK*.
5. From the Project window, the new Variable will display. Using your mouse *right click* the Variable name and select *Add Function > Lowercase* (or *Uppercase*).
6. The Uppercase (or Lowercase) Function will be added to the Current Project window. From the eFORMz Viewer, Form and Data Viewer, the Function change will display.



Cross-Reference: For more information on Lowercase and Uppercase, view the following tutorial: <http://support.minisoft.com/index.php/right-left-trim-video-only/>

URL Encoder

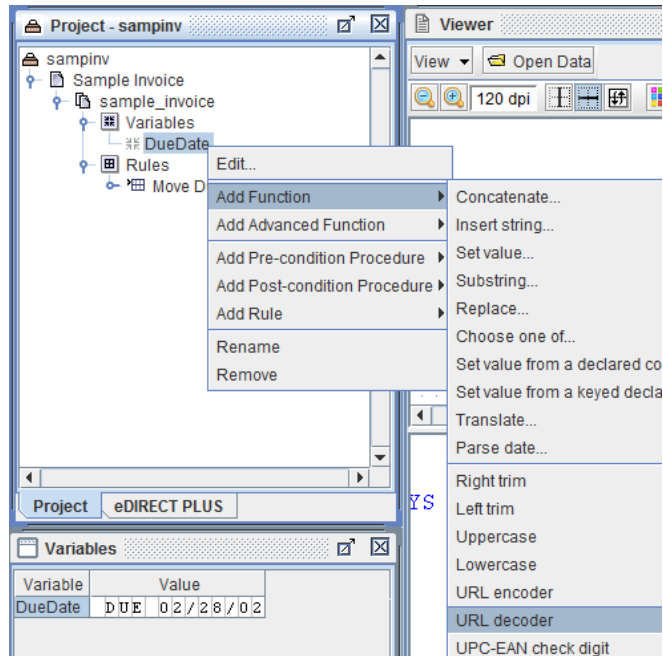
URL encoding converts the characters of a URL into a format that can be transmitted over the Internet. This is used in conjunction with the *Call a Web Service Procedure*:



Cross-Reference: For a tutorial on using the *URL Encoder* Function, see the *UPC-EAN Check Digit* part of the *Barcodes Mini-Manual*.

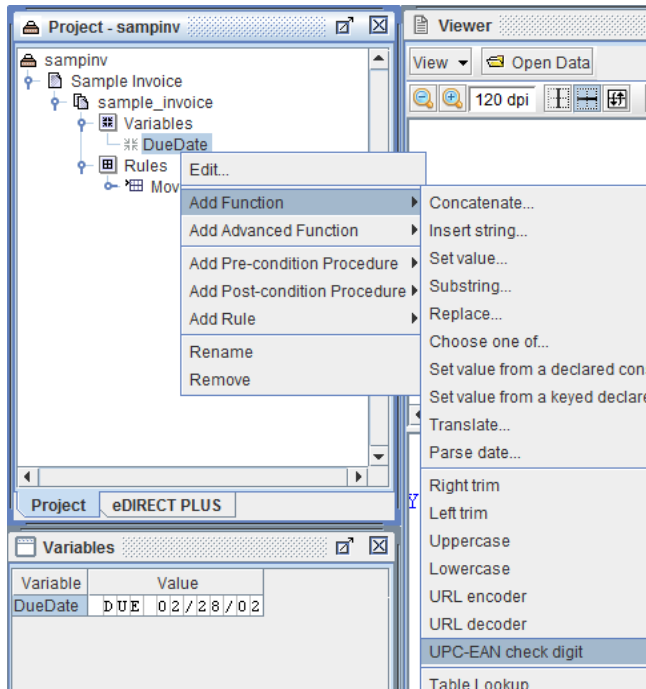
URL Decoder

URL Decoder decodes a string that has been URL-encoded. This is used in conjunction with the *Call a Web Service Procedure*:



UPC-EAN Check Digit

If the data for a UPC-A, UPC-E, EAN8 or an EAN13 barcode doesn't have a check digit, it can be calculated with the *UPC-EAN Check Digit* Function:



Cross-Reference: For a tutorial on using the *UPC-EAN Check Digit* Function, see the UPC-EAN Check Digit part of the *Barcodes Mini-Manual*.

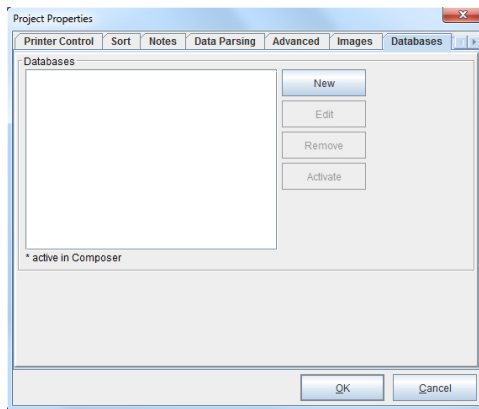
Cross-Reference: For more information on UPC-EAN Check Digit, view the following tutorial:

<http://support.minisoft.com/index.php/upc-ean-check-digit-video-only/>

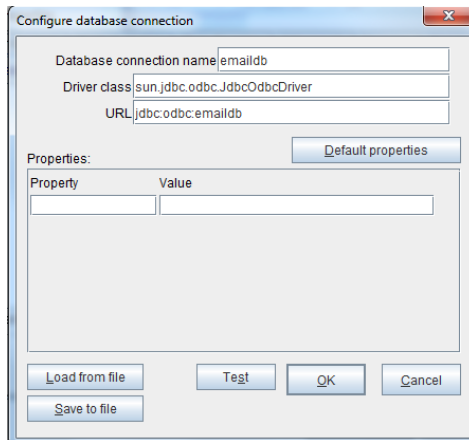
Table Lookup

To implement a database table lookup within the eFORMz Composer:

1. From the *Project Properties > Database* tab add a new database reference by selecting *New*.



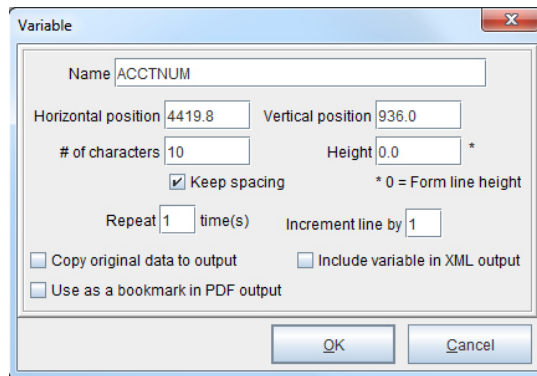
2. The *Configure Database Connection* dialog box displays. Enter the *database name*, *driver class*, and *url*.



NOTE: For further information on supported url and driver classes, see *Appendix B*.

SECTION 2

- Once the database has been configured select *Test* to confirm the connection. Once complete, click *OK*.
- The database name will display on the *Project Properties > Database* tab. Select *Activate* so that the information from the database becomes “real-time” data in eFORMz. Once complete, click *OK*.
- From the eFORMz Data Viewer, select the database key (for example, account number) that you would like to do a database lookup on. *Right click* and select *Add Variable*.
- The Variable dialog box will display. Enter a new Variable name and click *OK*.



Variable dialog box showing configuration for a new variable. The Name field is set to "ACCTNUM". Other fields include Horizontal position (4419.8), Vertical position (936.0), # of characters (10), Height (0.0), Repeat (1 time(s)), and Increment line by (1). There are three unchecked checkboxes: "Copy original data to output", "Include variable in XML output", and "Use as a bookmark in PDF output". The "Keep spacing" checkbox is checked. The "OK" and "Cancel" buttons are visible at the bottom.

- Once the new Variable displays in the Project window, *right click* the new Variable name and select *Add Function > Table Lookup*.
- The Table Lookup dialog box will display. From the Database drop down field select the database you would like to use.

Once a database has been selected the Table, Key, and Column fields will become available.

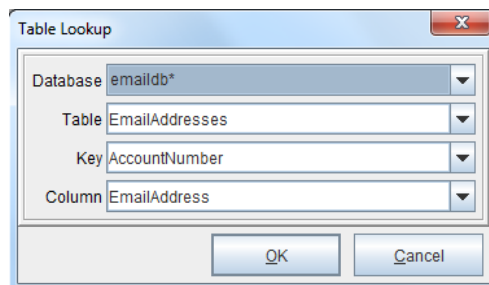


Table Lookup dialog box showing configuration for a table lookup. The Database field is set to "emaildb*", the Table field is set to "EmailAddresses", the Key field is set to "AccountNumber", and the Column field is set to "EmailAddress". The "OK" and "Cancel" buttons are visible at the bottom.

- Once the selections have been made, click *OK*.
- The Variable will now display the Table Lookup Function. A Rule can be added to the Variable with the database lookup by *right clicking* the Variable and selecting *Add Rule*.

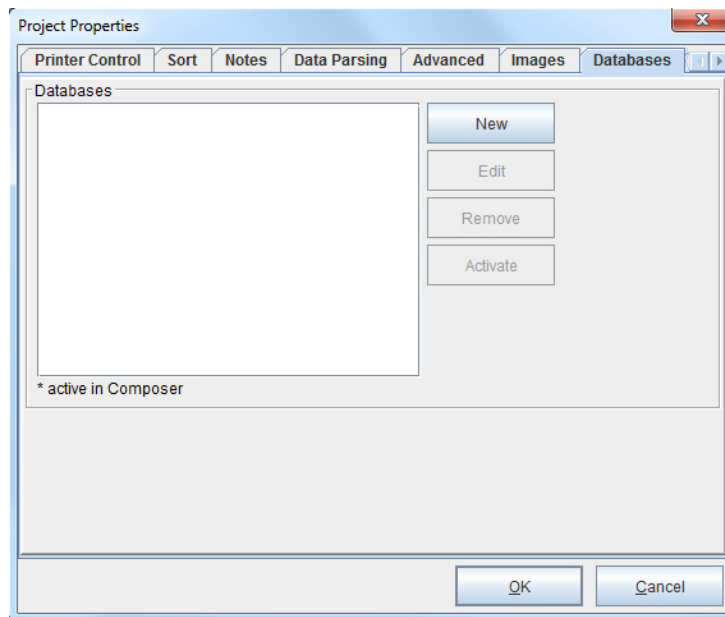
SQL Lookup

The purpose of an SQL Lookup is to return additional data to an eFORMz project based upon a value in a data file. Normally this value is an indexed item to a table in a database.

How to implement

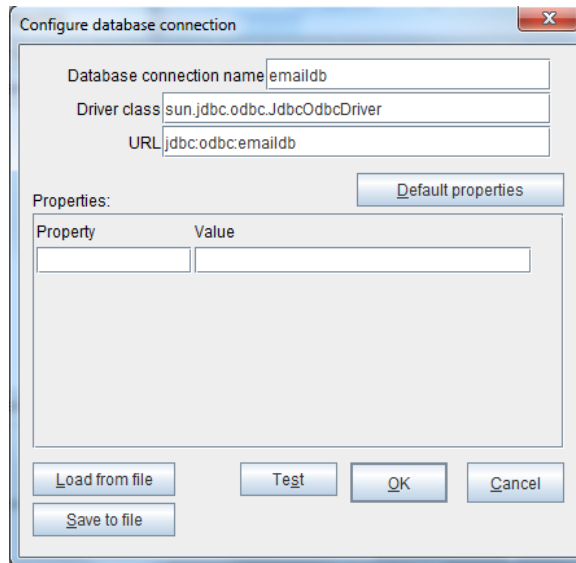
To implement an SQL Lookup:

1. If you haven't done so already, from the *Project Properties > Database* tab add a new database reference by selecting *New*.



SECTION 2

2. The *Configure Database Connection* dialog box displays. Enter the *database name*, *driver class*, and *url*.



NOTE: For further detailed information on supported url and driver classes, see *Appendix B*.

3. Once the database has been configured select *Test* to confirm the connection. Once complete, click *OK*.
4. The database name will display on the *Project Properties > Database* tab. Select *Activate* so that the information from the database becomes "real-time" data. Once complete, click *OK*.
5. Create a Variable containing the value you want to find in your database.
6. Add the Functions *Right Trim* and *Left Trim* to the Variable. This removes any possible leading and trailing spaces that would cause your lookup to fail.

TIP: To confirm the value, view the results in the Variables Viewer window.

7. Create a second Variable to be populated by the first item in your *SELECT* statement. If more than one column of data will be returned create those Variables now. From the Variables placeholder right click and select *Variables > By Position*.
8. Give your new Variable a name and then select *OK*.
9. Select your newly created Variable, *right-click*, and select *Function > SQL Lookup*.

eFORMz Composer Data Customization

10. The SQL Lookup dialog box appears. In the database dropdown box, select the database you want to access.
11. Enter the SQL Statement you want executed. Use "?" as replacement characters for parameters you want to pass to the statement.
12. Add parameters in the order you want the "?" replaced in your statement.
13. If you are returning more than one column of data, add the additional columns, selecting the Variable you want the returned value stored in the 'Additional columns assigned to' field.

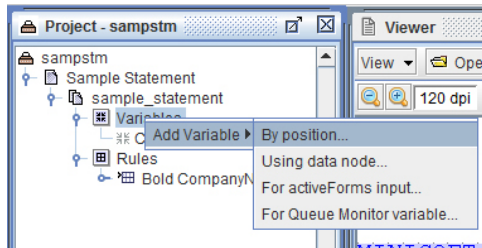
NOTE: For complex queries write a stored procedure and execute it. The stored procedure must return a result set, not just a value.

14. Once complete select *OK*.

Does File Exist

Add this Function to a Variable to determine whether a given file exists or does not exist.

1. To verify if a specific file exists, create a Variable *By position*.



2. Assign the Variable a name and select *OK*.
3. *Right click* the Variable and select *Add Function > Does file exist...*
4. Type in the name of the file.
5. The answer will display in the Variable Window.

Cross-Reference: For more information on determining if a file exists, view the following tutorial: <http://support.minisoft.com/index.php/does-file-exist-video-only/>

Advanced Functions

eVS Barcode Data Generator

The Electronic Verification System (eVS), also known as the USPS PRS barcode, allows high-volume package mailers and package consolidators to document and pay postage, including special service fees, using electronic manifest files. The files are transmitted over the Internet to a Postal Service™ database. eVS is designed to make it easy for high-volume package mailers to take advantage of destination entry rates.

eFORMz provides the ability to generate this barcode using either hard-coded 'constant' values and/or referencing data within the print file using Variable assignments.

In the example below, the following values are assigned accordingly:

- ◆ Application Identifier
- ◆ Service Type Code
- ◆ Customer ID or DUNS number
- ◆ Sequential Package ID

The screenshot shows a dialog box titled "eVS Barcode Data Generator". It contains four input fields arranged in a 2x2 grid. Each field has a radio button for "Constant" (which is selected) and a radio button for "Variable". The fields are: Application Identifier (91), Service Type Code (023), Customer ID or DUNS Number (123456789), and Sequential Package ID (98765432). Below these fields is a dropdown menu labeled "Output displayable barcode data to:" with "N/A" selected. At the bottom right are "OK" and "Cancel" buttons.

Application Identifier: The Application Identifier (AI) is a 2-digit code used within UCC/EAN Code 128 that identifies the barcode as a postal barcode. The AI values for Confirmation Services are 91 and 420, and appear as part of the human-readable printed data. If concatenation is used to include the postal routing code (destination ZIP Code), then both the AI of 91 and 420 must appear as part of the human-readable print for the routing barcode component. Neither Application Identifier 91 nor 420 may

eFORMz Composer Data Customization

appear in parentheses

Service Type Code: Service Type Codes (STCs) are three-digit designators included in the Confirmation Services barcode to identify the class of mail and/or type of Extra Services included on that specific mail item. See Table 1 in Appendix G for the Service Type Codes.

Customer ID/Dun & Bradstreet Number (DUNS): A customer ID number provided by the Postal Service or a DUNS number must be used in the barcode to identify the mailer or the mailer's client. A DUNS number uniquely identifies businesses at specific geographical locations. Mailers or their clients who need to obtain their DUNS may request it by contacting Dun & Bradstreet at (800) 333-0505 or via the Internet at <http://www.dnb.com>. Mailers with special ID requirements can contact the eVS program office via eVS@usps.gov.

Sequential Package ID: Customers assign a 7-digit Sequential Package Identifier. This string of numbers may be a Variable length field of 2 to 7 digits using the UCC/EAN Code 128 symbology.

Check Digit – MOD 10: The MOD 10 check digit validates the barcode authenticity. This value is automatically calculated using the barcode generator and is the last (single digit) value of the barcode.

SmartLabel Design and Barcode

SmartLabel is an intelligent pre-paid, pre-addressed return-shipping label integrated onto the order summary and shipped with the consumer's merchandise. SmartLabel's dynamic bar code links the package to the customer's original order and provides returns information links vital to both the customer and the retailer.

Transporter Code: This two-digit code indicates the front-end carrier and the service used to induct the package into the Newgistics return network. It occupies positions 2 and 3 within the SmartLabel barcode.

<i>Code</i>	<i>Description</i>
21	USPS First Class
25	USPS Return Delivery Unit

Service Code: This two-digit code the Newgistics service requested and the request mechanism. It occupies positions 4 and 5 of the SmartLabel barcode.

<i>Code</i>	<i>Description</i>
00	SmartLabel (in box generated by client)
01	Direct label (pre-printed)
10	CSR SmartLabel (generated by Shipment Manager)
20	Consumer SmartLabel (generated by Shipment Manager)

eFORMz Composer Data Customization

Origin ZIP: This is the 5-digit postal ZIP of the point of origin for the return.

Disposition Rule Set Name: This is the last two characters of the merchant specific Disposition Rule Set name used to determine the sort (disposition) of the package. The set name occupies positions 11-12.

Newgistics Merchant ID: Four character identifier assigned by Newgistics. This identifier may be alphanumeric and occupies position 13-16.

Client Service Flags: These flags indicate order (returns vs exchange) or charge status. They occupy either position 17.

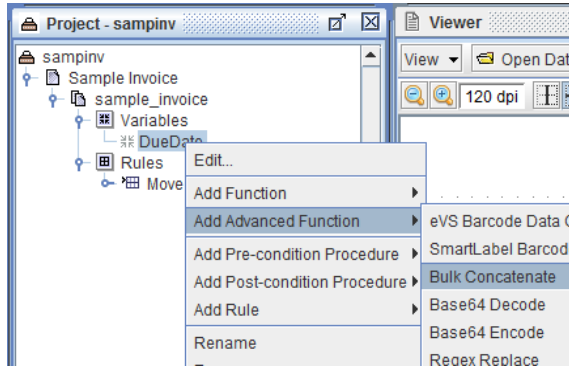
<i>Code</i>	<i>Description</i>
0	Standard
1	Complimentary
2	Exchange

Client Data Version: This is a version indicator for client-specified data. During your initial SmartLabel implementation, this value is set to 1. In subsequent revisions to the SmartLabel barcode, the field is incremented. This occupies position 18.

Client Data: This is serialized client data structure. This is composed of the mandatory Return Identifier fields and any other fields identified by your version. The length of this field will be fixed for any particular client version of the barcode. It occupies either position 19-47.

Bulk Concatenate

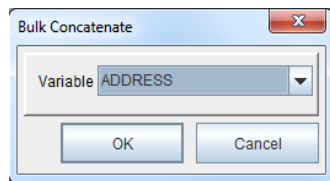
Bulk concatenation refers to the ability to search large sections of text or to display specified text in a string format.



How to implement

To implement a Bulk Concatenate Function:

1. From the Current Project window choose a Variable, *right click*, and select *Add Function > Bulk Concatenate*.
2. The *Bulk Concatenate* dialog box displays. Select a Variable, or choose from the range of System Variables.



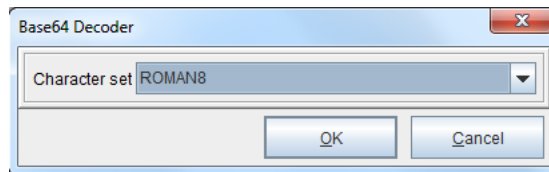
3. The chosen Variable will now display the *Bulk Concatenate* Function.
4. From the eFORMz Form Viewer, *right click* and select *Add Rule > Place Text*.

5. From the Rule Properties dialog box enter a new Rule name and click *OK*.
6. From the *Place Text Action* dialog box, keep the existing Horizontal and Vertical position and click *OK*.

Cross-Reference: For more information on using the Bulk Concatenate Function, view the tutorial: <http://support.minisoft.com/index.php/bulk-concatenate-video-only/>

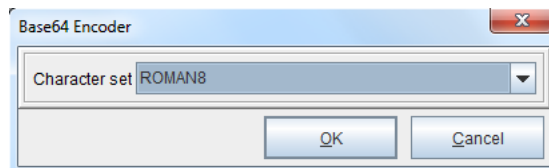
Base64 Decode

Base64 encodes binary data by treating it numerically and translating it into a base 64 representation. To decode data in a base 64 representation, use Base64 Decode.



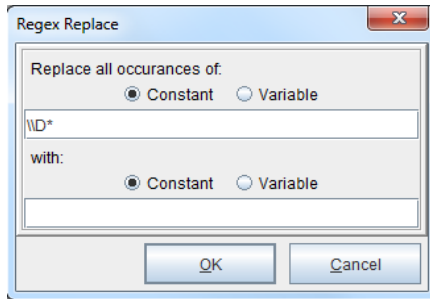
Base64 Encode

Base64 encodes binary data by treating it numerically and translating it into a base 64 representation. To encode data in a base 64 representation, use Base64 Encode.



Regex Replace

The *Regex Replace* Function in eFORMz allows you to search and replace parsed data in the eFORMz Composer using regular expressions.

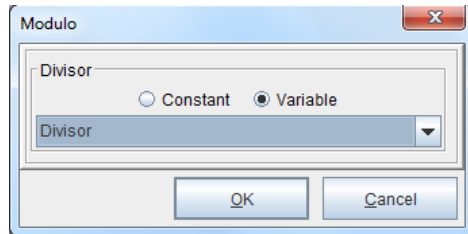


Cross-Reference: To learn more about regular expressions, consult the following entry on the Minisoft Support Wiki: <http://support.minisoft.com/index.php/regular-expressions-2/>

Cross-Reference: For more information on using the Regex Replace Function, view the tutorial: <http://support.minisoft.com/index.php/regex-replace-video-only/>

Modulo

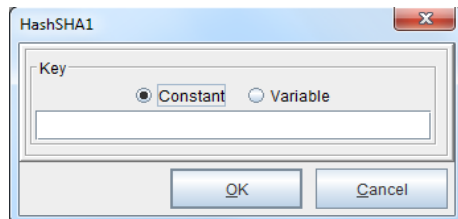
Use the *Modulo* Function to find the remainder after division of one number by another.



Cross-Reference: For more information on using the Modulo Function, view the tutorial: <http://support.minisoft.com/index.php/modulo-video-only/>

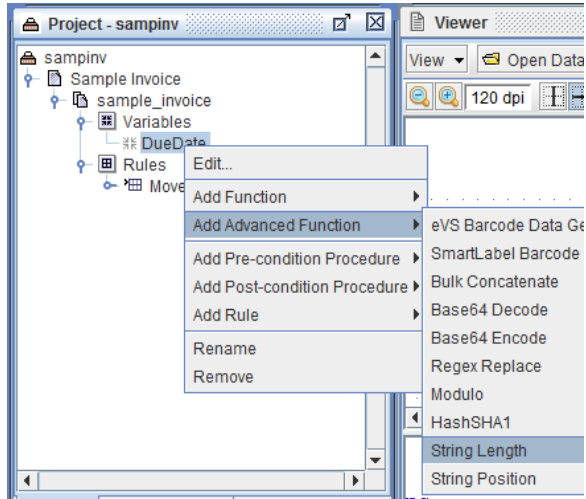
HashSHA1

SHA-1 (secure hash algorithm), is a cryptographic hash Function commonly used in security applications and protocols.



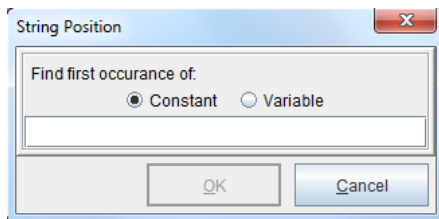
String Length

String Length determines the total length of a Variable's string of characters.



String Position

String Position determines the first occurrence of a character or a string of characters in a string.



Procedures

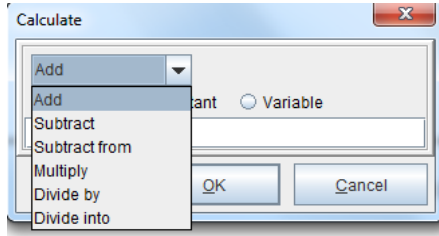
eFORMz supports the following Procedures:

- ◆ Calculation
- ◆ Set Variable
- ◆ Sum
- ◆ Date Calculation
- ◆ Sort
- ◆ Numeric Format
- ◆ Date Format
- ◆ Time Interval Format
- ◆ SQL Load
- ◆ SQL Update
- ◆ SQL End Transaction
- ◆ Load Variable from a Declared Constant
- ◆ Load Variable from a Keyed Declared Constant
- ◆ Load Variable from a Variable
- ◆ Load Variable from Project Output
- ◆ Load Variable from XML Node
- ◆ Load a Component
- ◆ Call a Web Service
- ◆ Call a Dynamic Service
- ◆ File List
- ◆ Printer List

By default, all procedures are 'Pre-condition.' These procedures are executed before the form condition is evaluated. 'Post-condition' procedures are only executed if the form condition evaluates to true.

Calculation

Calculation implements numeric math calculations.



Actions include:

- ◆ Add
- ◆ Subtract
- ◆ Subtract From
- ◆ Multiply
- ◆ Divide by
- ◆ Divide into

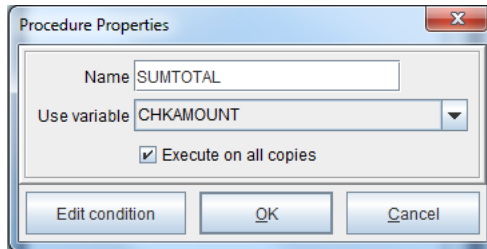
How to implement

The following example will demonstrate the use of the *Add* and *Sum* actions with math calculations. To create a math calculation:

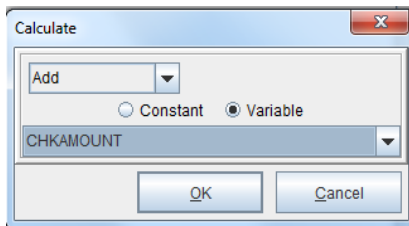
1. Create a variable that encompasses the value(s) you would like to calculate.
2. From the Current Project window *right click* the form you would like to add the Procedure to and select *Add Procedure > Calculation*.

eFORMz Composer Data Customization

3. The Procedure Properties dialog box will display. Enter a procedure name and then select the variable you created earlier. Select *OK*.



4. The Calculation dialog box displays. Select an action from the drop down box. Options available are *add, subtract, subtract from, multiply, divide by, divide into*.



5. To sum the calculation, create a Global Variable by *right clicking* the project name and selecting *Add Global Variable*. *Right click* the calculation procedure (SUMTOTAL) and select *Add Action > Sum*. From the Variable drop-down menu, select the Global Variable created earlier (GlobalVariable1). Click *OK*.
6. To place the calculation, *right click* the location in the eFORMz Form Viewer where it should be placed and select *Add Rule > Place Text*. From the Use variable drop-down menu, select the Global Variable (GlobalVariable1). Click *OK*. The value will display:

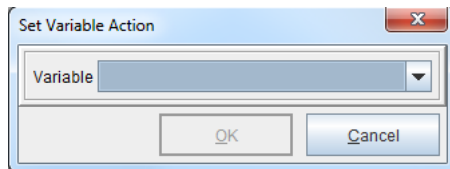
Cross-Reference: For more information on Calculation, view the following tutorial:

<http://support.minisoft.com/index.php/calculation-procedure-video-only/>

Set Variable

Set a variable using a procedure. For example, a procedure that is running a calculation, when completed, will place the total into another variable. Sum provides an ample illustration.

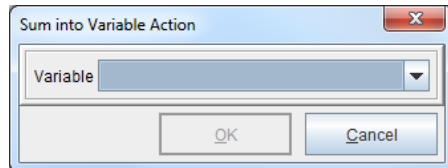
To set the value of a variable from a procedure, use *Set Variable*.



An example of this might be taking the results of a *Calculation*, and using the *Set Variable* Procedure, placing the total into another variable.

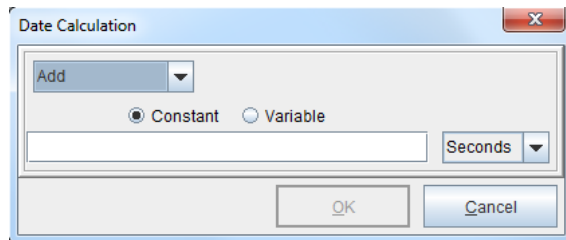
Sum

Sum a column of numbers using the following method: See step #5 from the Calculation tutorial.



Date Calculation

Date Calculation adds or subtracts a time interval (years, months, days, hours, minutes, seconds) to or from a timestamp.

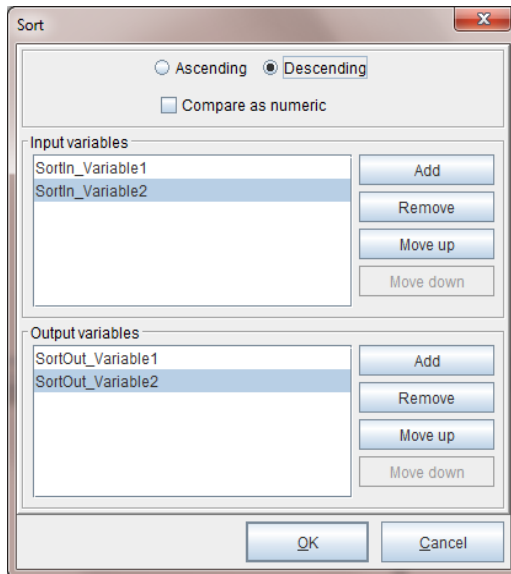


Cross-Reference: For more information on Date Calculation, view the following tutorial:

<http://support.minisoft.com/index.php/date-calculation-procedure-video-only/>

Sort

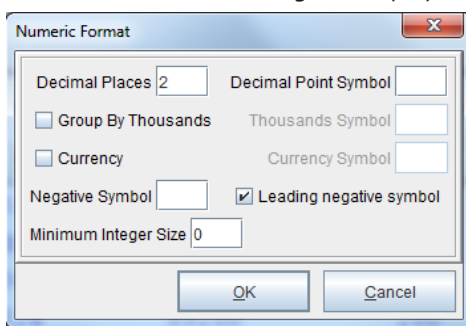
Multiple sorts, in either Ascending or Descending order, can be performed on Variable input values. The order is important as the values in the example below for 'SortIn_Variable1' will be sorted and placed into 'SortOut_Variable1'.



Numeric Format

Format a procedure to a numeric value using decimals, group separators, and currency symbols. To implement a numeric format:

1. From the Project window select the Procedure you would like to add the numeric format option to. Right click *Action* and select *Numeric Format*.
2. The Numeric Format dialog box displays. Specify your options and then select *OK*.



Options include:

Decimal Places

Determines the location of the decimal separator in the result string.

Decimal Point Symbol

Default is a (.) period. Customize your decimal point using this option.

Group By Thousands

Serves as both a group separator and a number scaling specifier. As a group separator, it inserts a localized group separator character between each group. As a number scaling specifier, it divides a number by 1000 for each comma specified.

Thousands Symbol

Default separator is a (,) comma. Customize your separator using this option.

Currency and Currency Symbol

Default currency sign (\$). Sets the numeric value to currency. Set a custom sign in the Currency Symbol field.

Negative Symbol and Leading negative symbol

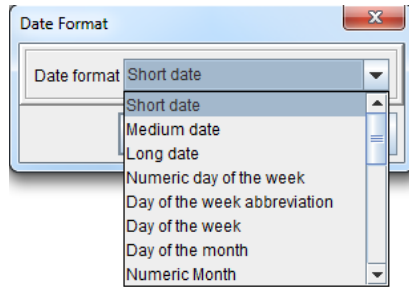
Default symbol (-). Sets the numeric value to negative by adding a negative symbol. To set the negative symbol at start of value select *Leading negative symbol*.

Minimum Integer Size

Placement holder for integer value.

Date Format

The *Date Format* feature converts a date from Epoch time (# of milliseconds since 1/1/1970). This is often used in conjunction with the Current Time (Timestamp) system variable, which returns the current time in the number of milliseconds since January 1, 1970.

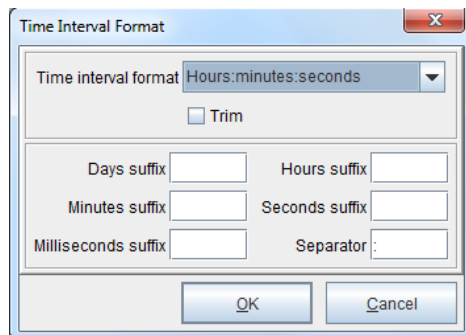


Format options include:

Short date, Medium date, Long date, Numeric day of the week, Day of the week abbreviation, Day of the week, Day of the month, Numeric Month, Month abbreviation, Month, Year, Short time, Medium time, Long time, Hour (24 hour format), Hour (12 hour format), Minute, Second, Millisecond, AM/PM, Hour (24 hour format), Short timezone, Long timezone, RFC822 timezone.

Time Interval Format

The *Time Interval Format* feature converts a time from Epoch time (# of milliseconds since 1/1/1970). This is often used in conjunction with the Current Time (Timestamp) system variable, which returns the current time in the number of milliseconds since January 1, 1970.



The screenshot shows a dialog box titled "Time Interval Format" with a close button (X) in the top right corner. The dialog contains the following elements:

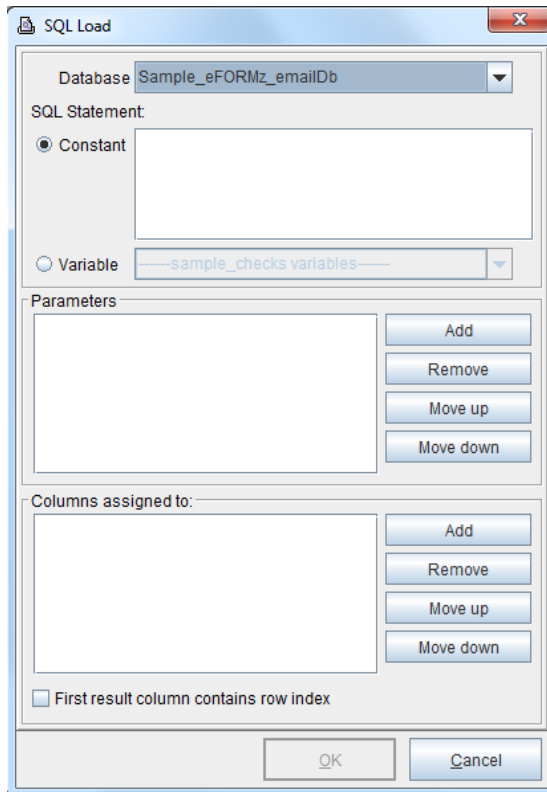
- A dropdown menu labeled "Time interval format" with the text "Hours:minutes:seconds" selected.
- A checkbox labeled "Trim" which is currently unchecked.
- Input fields for "Days suffix", "Hours suffix", "Minutes suffix", "Seconds suffix", and "Milliseconds suffix".
- An input field for "Separator" followed by a colon (:).
- "OK" and "Cancel" buttons at the bottom.

Format options include:

Hours:Minutes:seconds.milliseconds, Hours:minutes:seconds, Hours:minutes, Hours, Minutes:Seconds.milliseconds, Minutes:Seconds, Minutes, Seconds.ms, Seconds, milliseconds.

SQL Load

The *SQL Load* feature will load variables with values from an SQL query. This is particularly useful for returning arrays. The variables will occur the same number of times as there are rows returned from the query.



The screenshot shows the "SQL Load" dialog box. It has a title bar with a close button. The "Database" dropdown is set to "Sample_eFORMz_emailDb". Under "SQL Statement", the "Constant" radio button is selected, and there is an empty text area. The "Variable" radio button is unselected, and a dropdown menu shows "sample_checks variables". The "Parameters" section has an empty list box and buttons for "Add", "Remove", "Move up", and "Move down". The "Columns assigned to:" section also has an empty list box and buttons for "Add", "Remove", "Move up", and "Move down". At the bottom, there is a checkbox labeled "First result column contains row index" which is unchecked, and "OK" and "Cancel" buttons.

SQL Update

SQL Update will return a single record from an SQL database. For returning arrays, use *SQL Load*.

How to implement

To implement a database SQL update within the eFORMz Composer:

1. From the *Project Properties > Database* tab add a new database reference by selecting *New*.
2. The *Configure Database Connection* dialog box will display. Enter the *database name*, *driver class*, and *url*.

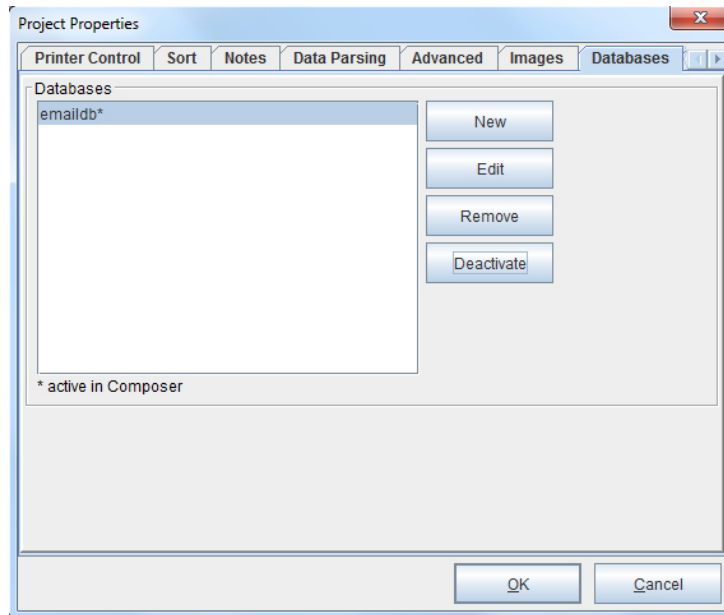
The screenshot shows a dialog box titled "Configure database connection". It contains three text input fields: "Database connection name" (value: emaildb), "Driver class" (value: sun.jdbc.odbc.JdbcOdbcDriver), and "URL" (value: jdbc:odbc:emaildb). Below these fields is a "Properties:" section with a "Default properties" button and a table with "Property" and "Value" columns. At the bottom are buttons for "Load from file", "Save to file", "Test", "OK", and "Cancel".

NOTE: For further detailed information on supported url and driver classes, see *Appendix B*.

3. Once the database has been configured select *Test* to confirm the connection and then click *OK*.

SECTION 2

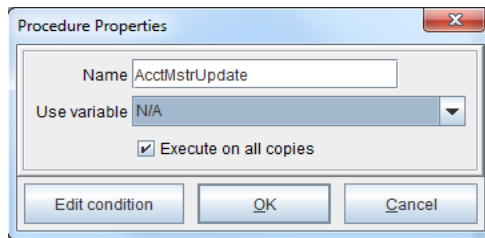
4. The database name will display on the *Project Properties > Database* tab. Select *Activate* so that the information from the database becomes "real-time" data in the eFORMz Viewer, Form and Data Viewer.



Once complete, click *OK*.

To create the *SQL Update* statement create a new procedure. Note that the new procedure does not need to be attached to a variable:

1. From the Project window *right click* the form and select *Add Procedure > SQL Update*.
2. From the *Procedure Properties* dialog box enter a procedure name and select a variable or N/A for a variable option. Click *OK*.



3. The *SQL Update Action* dialog box appears:

- From within the SQL Update dialog box, enter the appropriate SQL statement to either UPDATE, DELETE or INSERT values within your database. In the example above, each question mark (?) is represent by the sequential list of parameters associated with eFORMz variables.

SQL Parameter		eFORMz Variable
statusflag	=	StatusFlg
updatedate	=	Eloquence_CurrDT
accountnumber	=	AccountNumber

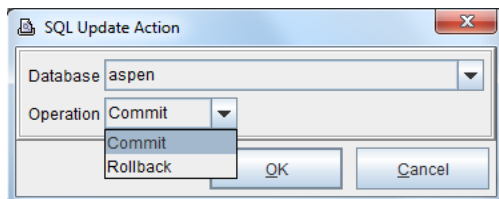
New parameter-variable assignments can be made by clicking on the *Add* or *Delete* button. To delete a parameter-variable simply select the variable you would like to remove from the list and click *Remove*. There is also the option to rearrange the order by using the *Move up* or *Move down* button.

NOTE: Special caution is advised when specifying a select statement to update a database. Once the database is defined and information is entered in the statement field, real time updates will occur if a data file is processed. This will reflect in immediate changes to your database. Only persons familiar with database design should be permitted to enter this information into the statement field. Minisoft will not be held responsible for information entered resulting in database changes.

- Once complete, click *OK*.

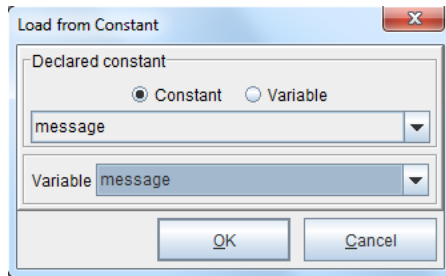
SQL End Transaction

The *SQL EndTransaction* feature is used in conjunction with the *SQL Update Procedure* to set *AutoCommit* to off. There are two options: *commit* and *rollback*. *Commit* marks the end of a successful transaction by making all data modifications performed since the start of the transaction a permanent part of the database. *Rollback* rolls back a transaction to the beginning in order to erase all data modifications made from the start of the transaction.



Load Variable from a Declared Constant

This procedure loads a variable from a constant declared in the eFORMzCfg_User.xml file. The repeat of the variable will be set to the number of values in the declaration. For example:



```
<Constant Name="message">
  <![CDATA[Shipping and Delivery]]>
  <![CDATA[Request Order Changes]]>
  <![CDATA>Returns and Replacements]]>
  <![CDATA[Payment and Billing]]>
  <![CDATA[Product/General Questions]]>
  <![CDATA[Site Help]]>
</Constant>
```

The repeat of the variable will be set to 6.

Cross-Reference: For a detailed example, see the *Language Substitution* Part of the *Document Template Mini-Manual*.

Load Variable from a Keyed Declared Constant

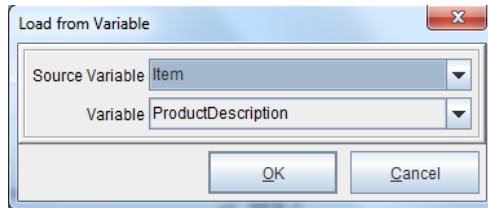
This procedure loads a variable from a keyed constant declared in the eFORMzCfg_User.xml file. For example:

```
<KeyedConstant Name="OrderNbrLabels">
  <Entry Key="EN">
    <![CDATA[Order Number]]>
  </Entry>
  <Entry Key="FR">
    <![CDATA[French Order Nbr]]>
  </Entry>
  <Entry Key="GE">
    <![CDATA[German Order Number]]>
  </Entry>
</KeyedConstant>
```

Cross-Reference: For a detailed example, see the *Language Substitution* Part of the *Document Template Mini-Manual*.

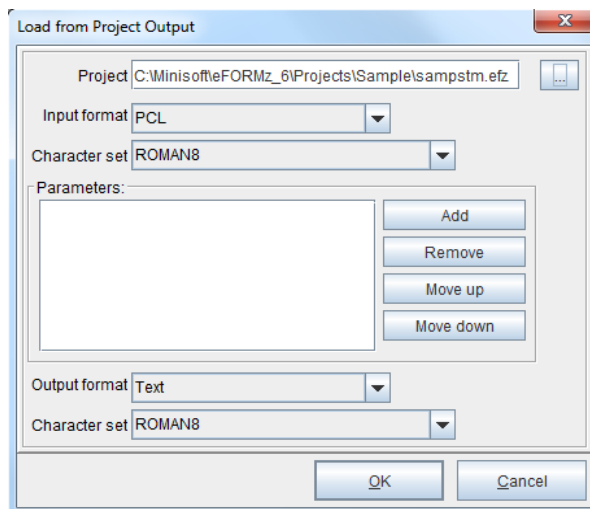
Load Variable from a Variable

This procedure loads a variable with the values of another variable. The repeat of the destination variable will be set to number of values of the source variable.



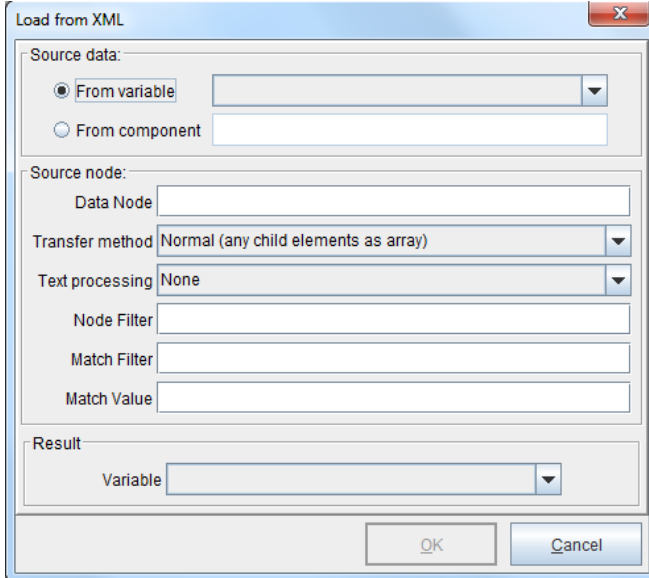
Load Variable from Project Output

This procedure is useful for retrieving text or XML output from another project and placing it into a variable.



Load Variable from XML Node

This procedure is useful for loading a Variable from XML in another Variable, using the same techniques as loading a Variable from an XML data file. This is useful when needing information from XML data returned from a database query or web service call.



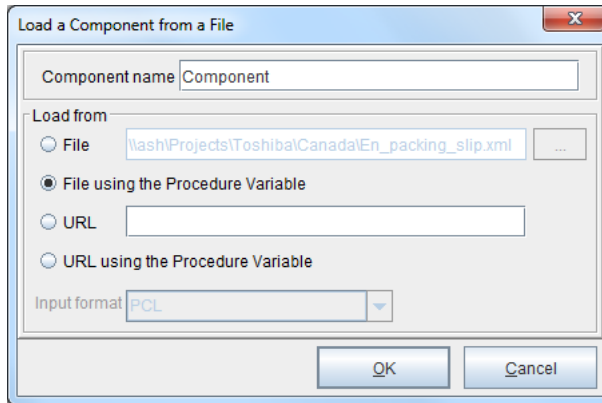
The screenshot shows a dialog box titled "Load from XML" with a close button (X) in the top right corner. The dialog is divided into several sections:

- Source data:** Contains two radio buttons. The first, "From variable", is selected and has an empty text box and a dropdown arrow next to it. The second, "From component", has an empty text box.
- Source node:** Contains a "Data Node" text box, a "Transfer method" dropdown menu set to "Normal (any child elements as array)", a "Text processing" dropdown menu set to "None", and three empty text boxes for "Node Filter", "Match Filter", and "Match Value".
- Result:** Contains a "Variable" dropdown menu with an empty text box and a dropdown arrow.

At the bottom of the dialog are two buttons: "OK" and "Cancel".

Load a Component

This procedure is useful for loading a component from a file or a URL.



Cross-Reference: For a use of Load a Component, follow along with the following tutorial:

<http://support.minisoft.com/index.php/creating-a-two-sided-label-w-a-reference-number-parts-i-and-ii/>

Call a Web Service

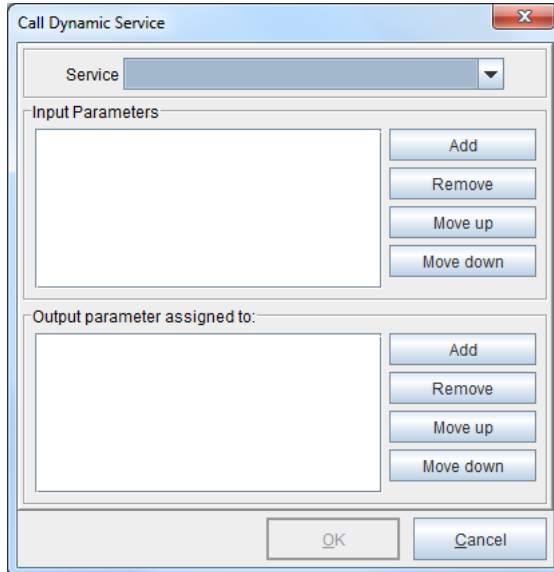
This procedure is used for calling and retrieving data from an 'http' accessible web service. The URL can be a constant string or a variable created earlier in the project. The parameters are submitted with either a GET or POST request. The data can be in any combination of URL Encoded or MIME input and output. The data returned from the web service is parsed for return values in the form "tag=value". The current value of the procedure is set to the return code of the request.

The screenshot shows the 'Call a Web Service' dialog box. At the top, there is a title bar with a close button. Below it, the 'Request verb' is set to 'POST' (radio button selected). The 'URL' section has 'Constant' selected, with an empty text box. The 'Variable' option is also present with a dropdown menu showing 'PackingList variables'. Below the URL section are two buttons: 'Request Headers' and 'Response Headers'. The 'Input Data' section has 'URL Encoded' selected. Below this is the 'Input Parameters' section, which is currently empty, with 'Add', 'Edit', and 'Remove' buttons to its right. The 'Output Data' section has 'URL Encoded' selected. Below this is the 'Output Parameters' section, also empty, with 'Add', 'Edit', and 'Remove' buttons to its right. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

Cross-Reference: For a tutorial on implementing Call a Web Service, see the *Web Services Mini-Manual*.

Call a Dynamic Service

This procedure is used for calling custom code running under the Dynamic Service.



File List

File List will display files from an archive.

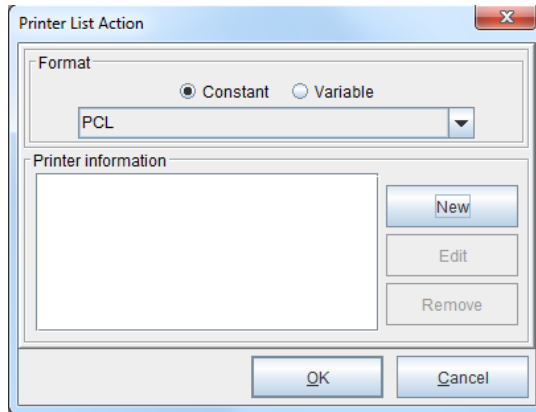
The screenshot shows a dialog box titled "Document List Action" with a close button (X) in the top right corner. The dialog is divided into several sections:

- Archive:** Contains radio buttons for "Constant" and "Variable" (selected). Below is a dropdown menu showing "Archive".
- Folder:** Contains radio buttons for "Constant" and "Variable" (selected). Below is a dropdown menu showing "CurrentFolder".
- Qualifier:** Contains radio buttons for "Constant" and "Variable" (selected). Below is a dropdown menu showing "FileMask".
- Search Operation:** A dropdown menu showing "Files".
- File information:** A list box containing:
 - File name --> DisplayFileName
 - Type --> DisplayFileType
 - Relative Name --> RelativeFileNameTo the right of the list box are three buttons: "New", "Edit", and "Remove".

At the bottom of the dialog are two buttons: "OK" and "Cancel".

Printer List

Printer List will display a list of printers in a variable.



Rule Actions

Once a variable has been created a *Rule Action* can be implemented to transform the look of your output and enhance its function or appearance.

Available rule actions include:

- ◆ Place Text
- ◆ 1D Barcode
- ◆ 2D Barcode
- ◆ Special Fonts
- ◆ Load Image
- ◆ Bar Chart
- ◆ Gauge Chart
- ◆ Page Number
- ◆ Translate
- ◆ Notify
- ◆ Output from Project

Cross-Reference: For information on 1D Barcodes, 2D Barcodes and Special Fonts, consult the *Barcodes Mini-Manual*. For more information on Bar Charts and Gauge Charts, refer to the *Graphical Charting Mini-Manual*.

Place Text Action

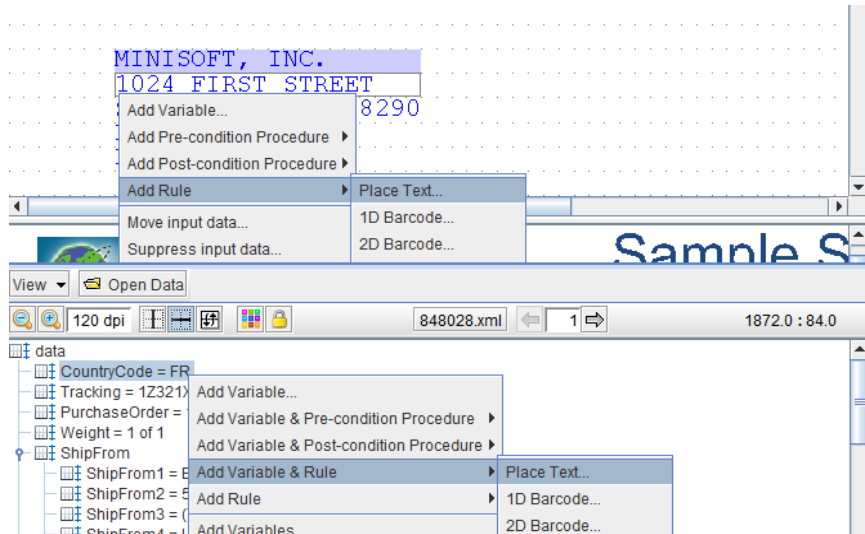
Place Text allows variables, custom or system, to be placed in a specified location on your output. Place Text actions can be created using multiple methods:

- ◆ *Right clicking* a variable from the Project Window and selecting *Add Rule > Place Text*.
- ◆ Direct Reference, which bypasses variables and directly places a particular string of characters. *Right click* data from the eFORMz Data Viewer and select *Add Rule > Place Text*.
- ◆ Scrolling to the location in the eFORMz Form Viewer where the text should appear, *Right clicking* and selecting *Add Rule > Place Text*.
- ◆ Selecting a string of characters in the eFORMz Data Viewer (*left click + drag*), *right clicking* and selecting *Add Rule > Place Text*.

How to implement

The following example shows the fourth method listed above.

1. From the eFORMz Data Viewer, select the data to copy or move (*left click + drag*). *Right click* and select *Add Rule > Place Text*. If you're using XML data, *right click* the XML node in the eFORMz Data Viewer, and select *Add Variable & Rule > Place Text*.

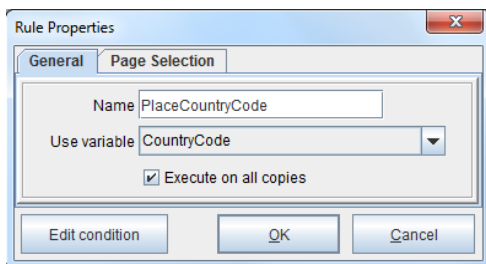


eFORMz Composer Data Customization

- The *Variable* dialog box will display. Enter a new name. Once complete, click *OK*.

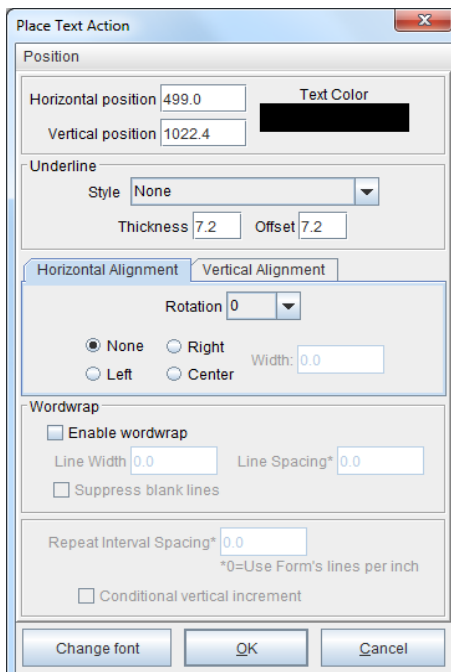
TIP: For a non-XML Variable, selecting *Copy original data to output* leaves the original data in place in the output window (bottom pane of the eFORMz Viewer). Not selecting this option removes the original placement in the output window and places the selected text in the position supplied in the Place Text dialog box.

- From the *Rules Properties* dialog box enter a new name and click *OK*.



The screenshot shows the 'Rule Properties' dialog box with the 'General' tab selected. The 'Name' field contains 'PlaceCountryCode'. The 'Use variable' dropdown menu is set to 'CountryCode'. The 'Execute on all copies' checkbox is checked. At the bottom, there are three buttons: 'Edit condition', 'OK', and 'Cancel'.

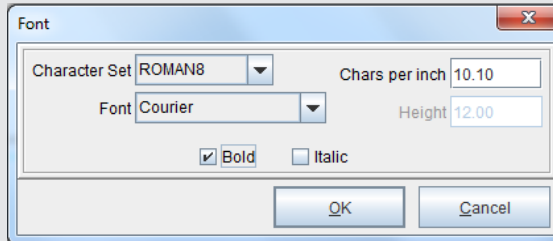
- The Place Text dialog box will display. If a position was selected earlier, select *Position > Paste*. If not, keep the existing horizontal and vertical positions - exact placement can be edited from the eFORMz Viewer later.



The screenshot shows the 'Place Text Action' dialog box. The 'Position' section has 'Horizontal position' set to 499.0 and 'Vertical position' set to 1022.4. The 'Text Color' is represented by a black square. The 'Underline' section has 'Style' set to 'None', 'Thickness' set to 7.2, and 'Offset' set to 7.2. The 'Horizontal Alignment' and 'Vertical Alignment' tabs are visible. The 'Horizontal Alignment' section has 'Rotation' set to 0, and radio buttons for 'None', 'Right', 'Left', and 'Center'. The 'Width' field is set to 0.0. The 'Wordwrap' section has 'Enable wordwrap' unchecked, 'Line Width' set to 0.0, 'Line Spacing*' set to 0.0, and 'Suppress blank lines' unchecked. The 'Repeat Interval Spacing*' field is set to 0.0, with a note '*0=Use Form's lines per inch'. The 'Conditional vertical increment' checkbox is unchecked. At the bottom, there are three buttons: 'Change font', 'OK', and 'Cancel'.

SECTION 2

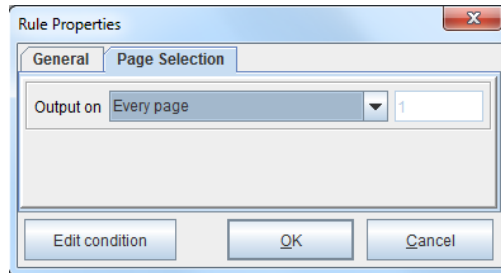
TIP: If you would like to change the font style for the copied selection, select *Change Font*. The Font properties dialog box will display.



6. Once complete, click *OK* from the *Place Text Action* dialog box.
7. The copied data will now appear in the bottom pane of the Viewer. To place the copied data in the desired position, using your mouse, drag and drop the data into place in the Form Viewer.

Page Selection

To specify what page(s) of output the Rule will display on:



Options include *First Page*, *Last Page*, *nth page*, *Every Page*, *First through nth page*, *nth through last page*, *First through nth to last page*, and *nth to last through last page* followed by the specified page number.

For example:

- ◆ "First through nth page" where $nth = 5$, the Rule will display on the first five pages.
- ◆ "nth through last page" where $nth = 5$, all pages except the first 4 will display the Rule.
- ◆ "First through nth to last page" where $nth = 5$, all pages except the last 5 will display the Rule.
- ◆ "nth to last through last page", where $nth = 5$, the Rule will display on the last 6 pages.

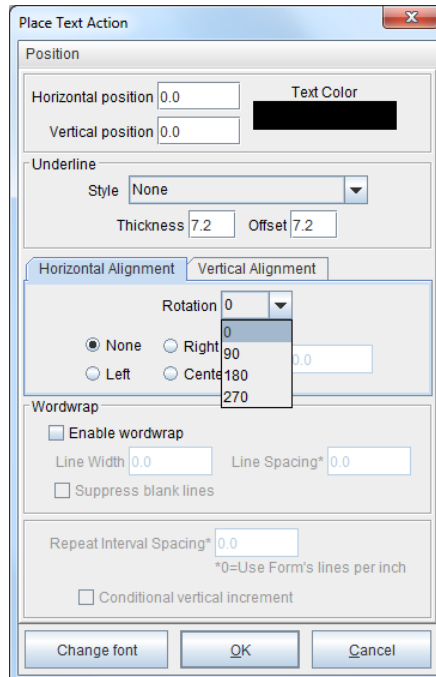
SECTION 2

Text Rotation

To rotate the characters used for a Place Text action:

1. From the *Place Text Action* dialog box select the rotation angle from the *Alignment > Rotation* field. Options available:

0, 90, 180, or 270 degrees



The image shows a screenshot of the "Place Text Action" dialog box. The "Rotation" dropdown menu is open, showing options for 0, 90, 180, and 270 degrees. The "None" radio button is selected under the "Horizontal Alignment" section. The "Text Color" field is currently black. The "Repeat Interval Spacing*" field is set to 0.0, with a note below it stating "*0=Use Form's lines per inch".

Field	Value
Horizontal position	0.0
Vertical position	0.0
Text Color	Black
Underline Style	None
Underline Thickness	7.2
Underline Offset	7.2
Rotation	0
Horizontal Alignment	None
Vertical Alignment	Top
Enable wordwrap	<input type="checkbox"/>
Line Width	0.0
Line Spacing*	0.0
Suppress blank lines	<input type="checkbox"/>
Repeat Interval Spacing*	0.0
Conditional vertical increment	<input type="checkbox"/>

Text Word Wrapping

To enable line wrapping, check the *Enable wordwrap* box. Options available:

Line Width - Specifies the wordwrap width in decipoints (720 dec = 1in).

Line Spacing and Repeat Interval Spacing - Specifies the distance between lines. Specify zero (0) if you would like to use the default form's lines per inch.

Conditional vertical increment - When set, the vertical coordinates for actions using repeating variables only gets incremented when the action is actually executed.

Suppress blank lines - When set, it removes space in between wrapped lines. Default is unchecked.

Place Text Action

Position

Horizontal position 138.0 Text Color [Black]

Vertical position 264.0

Underline

Style None

Thickness 7.2 Offset 7.2

Horizontal Alignment Vertical Alignment

Rotation 0

None Right Left Center Width: 0.0

Wordwrap

Enable wordwrap

Line Width 1000.0 Line Spacing* 10.0

Suppress blank lines

Repeat Interval Spacing* 0.0
*0=Use Form's lines per inch

Conditional vertical increment

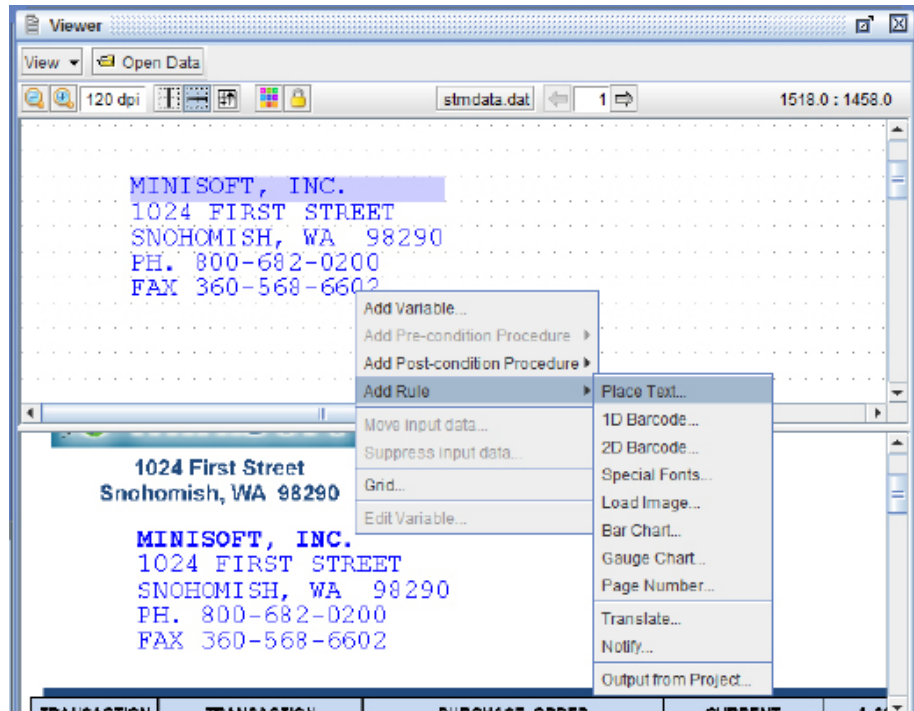
Change font OK Cancel

TIP: The *Append newline* feature found in the Concatenate Function options will only work in conjunction with the Wordwrap option.

Place Text Direct Reference

Placing text can be done without the creation of variables. This method is similar to selecting data from the Data Viewer and creating a variable, as a direct reference places a given string of characters in a specified location of the output. The following is an illustration.

1. From the eFORMz Data Viewer, *right click* the string of characters that are to be placed on the form and select *Add Rule > Place Text*.



eFORMz Composer Data Customization

2. The Rule Properties dialog box will display. Enter a value for the *Name* field and click *OK* to proceed.

Rule Properties

Name PlaceFAX

Horizontal position 0.0 Vertical position 1483.2

Offset 499.0 Height 93.4 *

Rotation 0 * 0 = Form line height

Repeat 1 time(s) Increment line by 1

Execute on all copies

Edit condition OK Cancel

3. The Place Text Action dialog box will display. Click *OK* to continue.
4. The string of characters will be placed according to its location in the Data Viewer, but like a variable, it can be dragged and dropped.

Cross-Reference: For a tutorial that utilizes placing input strings directly, view the "Place a Barcode from Input" video here:

<http://support.minisoft.com/index.php/place-a-barcode-from-input-video-only/>

Load Image

Place a .jpg, .gif, .bmp, .tif, .prn, .pdf, .png, or .frm image file using the *Load Image* action. Conditions can be used to allow the image to only appear when called for. Images can be loaded in one of seven ways:

- ◆ *Project* loads a previously saved image that was imported into the project using the *Project Properties > Images* option. Images stored in the project will cause larger project file sizes. View a tutorial here:

<http://support.minisoft.com/index.php/adding-an-image-from-a-project-video-only/>

- ◆ *Project using Rule Variable* loads a specified image from the *Project Properties > Images* based on a variable.
- ◆ *File* loads an image from a specified folder. The image will not be stored in the project file.
- ◆ *File using Rule Variable* loads a specified image from a folder based on a specified variable. View a tutorial here:

<http://support.minisoft.com/index.php/adding-an-image-from-file-using-a-variable/>

- ◆ *PDF file* loads a pdf file from a specified folder. The image will not be stored in the project file.
- ◆ *PDF file using the Rule Variable* loads a pdf file from a folder based on a specified variable.
- ◆ *Component* loads a .png, .jpg, .gif, .bmp, .tif or .pdf file from a composite data file.

Options include:

Desired width and height are configured in decipoints (720 decipoints = 1 in). Set the width and height for the *shrink to desired size* or *fit to desired size* options.

Scaling options include *None*, *Shrink to desired size*, *Fit to desired size*.

None leaves the size as is. *Shrink to desired size* will reduce the size of the image if it does not fit in the desired width and height specified. *Fit to desired size* will change the size of the image to match the desired width and height specified. The *Lock aspect ratio* option, when true, will keep the ratio of the images dimensions the same when scaling. This option figures which ratio (desired width / image width or desired height / image height) is needed for scaling both dimensions of the image to make it shrink or fit within the desired rectangle.

Cropping is enabled for images from a file. Images loaded from URLs or loaded directly into the project cannot be cropped.

Lock aspect ratio will keep the ratio of the image dimensions when scaling.

Conditional Vertical Increment when set, the vertical coordinate for actions using

eFORMz Composer Data Customization

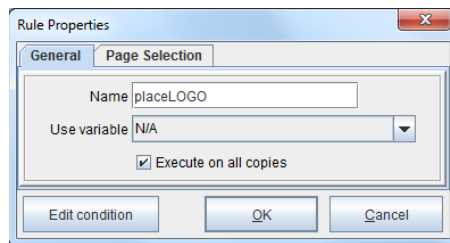
repeating variables only gets incremented when the action is actually executed.

Default resolution sets the resolution of the incoming image file.

How to implement

To load an image into an eFORMz project:

1. From the eFORMz Viewer, Form and Data Viewer, *right click* near the position you would like the image to appear. Select *Add Rule > Load Image*:
2. The Rule Properties dialog box will appear. From the *Use variable* drop-down menu, select the *N/A System Variable*.



SECTION 2

3. The Image Element dialog box will display. Select *Load Image from Project* or *Load Image from File*. For this example we will be loading an Image from a file.

Load Image Action

Position

Horizontal position 210.0 Repeat Interval Spacing* 0.0
Vertical position 120.0 *0=Use Form's lines per inch
Rotation 0 Conditional vertical increment

Scaling

Desired Width 0.0 Desired Height 0.0 Default resolution 0
Scaling None Lock aspect ratio

Cropping

X 0.0 Y 0.0 Width 0.0 Height 0.0

Load from

Project Logo_monochrome.jpg
 Project using the Rule Variable
 File ...
 File using the Rule Variable
 PDF File ...
 PDF File using the Rule Variable Load Page 1
 Component Component
Format png

OK Cancel

To specify the image from a file folder select the browse (...) button.

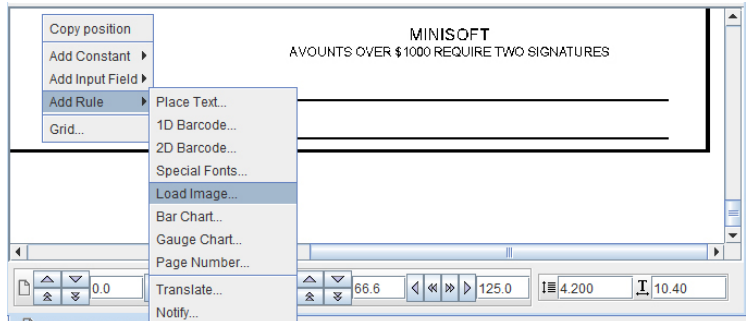
4. Once complete, click *OK*.

TIP: If the image doesn't appear in the exact location you would like it to appear, use your mouse to drag and drop it into position.

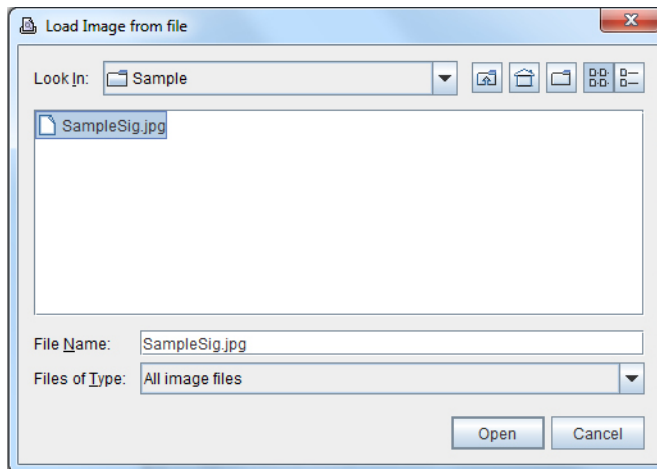
Load Image Based on a Condition

In the following example (SAMPCHK.efz) a signature will be added based on the contents of the data file. For example, if the check number appears in the data page, a signature will be applied.

1. *Right click* and select *Add Rule > Load Image* from the eFORMz Form Viewer.



2. From the Rule Properties dialog box enter a new rule name and select the *N/A System Variable*. Click *OK*.
3. From the Load Image Action dialog box select *Load Image from Project* or *Load Image from File*.
4. If using *Load Image from File*, use the browser button (...) to select the signature image you would like to appear.



5. Once the image has been selected, the path specifications will display in the Load Image Action dialog box. Adjust the scaling as necessary, and make any other needed modifications. Once complete, click *OK*.

SECTION 2

6. Create a condition for the new Load Image action that applies the signature only when a check number is present in the data file.
7. From the Project Window *right click* the rule for the image (LoadImage) and select *Add AND Condition > Is greater than*.
8. The Data Condition dialog box will display. From the *Use variable* field select the check number variable (CHK#). In the Constant field, enter an amount to be compared to.

The screenshot shows the 'Data Condition' dialog box. It features a title bar with a close button. The main area contains a dropdown menu with 'nth' selected and 'occurrence' to its right. Below this is a field labeled 'of variable' with 'CHK#' selected. A second dropdown menu shows 'Is greater than'. There are two radio buttons: 'Constant' (selected) and 'Variable'. Below the radio buttons is a text input field containing '0'. At the bottom left is a checkbox labeled 'Compare as Numeric'. At the bottom right are 'OK' and 'Cancel' buttons.

Once complete, click *OK*.

9. The image will only appear on pages containing a check number.

Cross-Reference: For various methods of conditionally placing images, view the "Supporting Multiple Brands with eFORMz" video here:

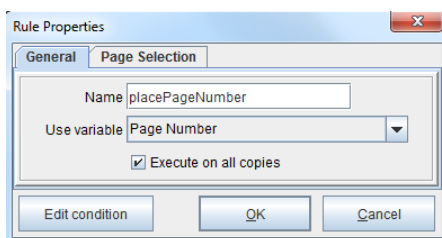
<http://support.minisoft.com/index.php/supporting-multiple-brands-video-only/>

Page Number

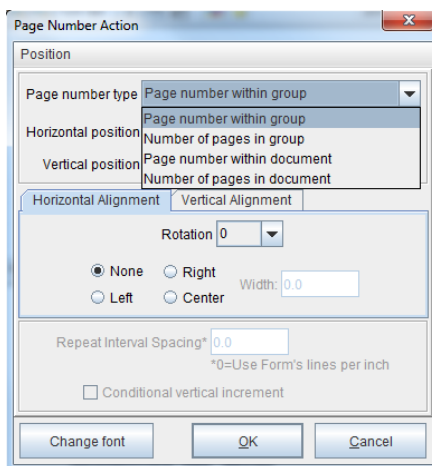
Place the page number of a form based upon the number of pages it contains or its location within a series of pages. Options include: *Page number within group*, *Number of pages in group*, *Page number within document* and *Page number in document*. The *Starting number* sets where the page count begins.

How to implement

1. *Right click* in the Form and Data Viewer and select *Add Rule > Page Number*.
2. The Rule Properties dialog box will appear. Enter a new rule name and select the System Variable *Page Number*.



3. The Page Number Action dialog box will display. Choose the desired option from the *Page number type* field. Once complete, click *OK*.



Translate

How to implement

To translate or convert data in a selected range:

1. From the eFORMz Viewer, *Input Data Viewer*, highlight the selected range that contains the data you would like to translate. For example, translate the forward slash (/) contained in the date to dashes (-):

Change:

05/01/00

To:

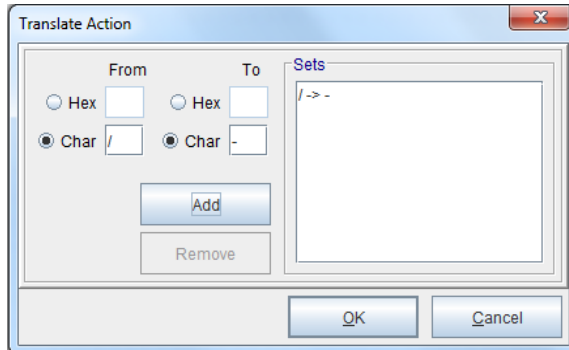
05-01-00

2. *Right click* and select *Add Rule > Translate*.
3. The Variable dialog box will display. Enter a new variable name and click *OK*.

4. The Rule Properties dialog box will display. Enter a new rule name and click *OK*.

eFORMz Composer Data Customization

- From the Translate dialog box, select *Char* for character change and then enter the translation in the *From* and *To* fields. When complete, click *Add* to add the translation set to the *Sets* field:



Once complete, click *OK*.

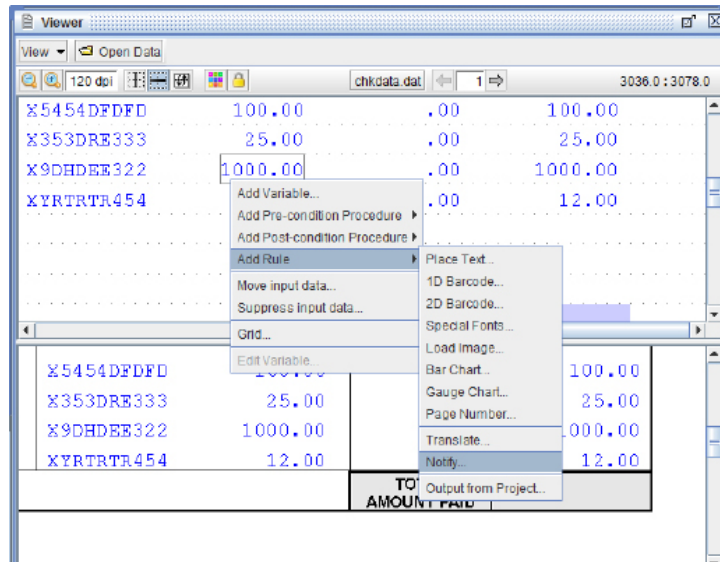
- To view the newly formatted data in the eFORMz Viewer, create a *Place Text* rule by selecting the same rule created for Translate (*DATE Translate*). *Right click* the *DATE Translate* rule and select *Add Action > Place Text*.
- The *Place Text* dialog box will display. In the horizontal and vertical position fields, the original position of the translated data will appear.
- Once complete, click *OK*. The translation will display in the eFORMz Form Viewer. To change the position, drag and drop the data in the eFORMz Form Viewer.

Notify

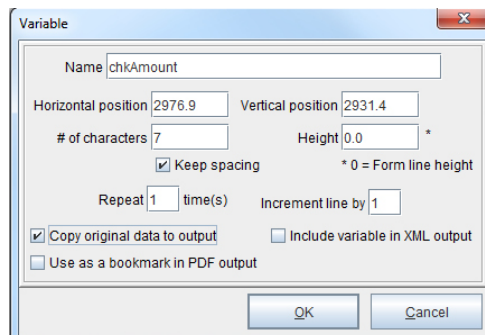
Setup a rule action to notify an individual or multiple individuals via email when a process is ran. The *Notify* action can be conditionally based.

How to implement

1. From the Data Viewer select the string of characters you would like to monitor, *right click*, and select *Add Rule > Notify*.



2. The Variable dialog box will display. Enter a new variable name. To keep the existing data in place select *Copy original data to output*. Once complete, click *OK*.



eFORMz Composer Data Customization

3. The Rules Properties dialog box displays. Enter a new rule name and click *OK*.
4. From the Notify Action dialog box enter the necessary information:

SMTP Host is your email server.

User is the username to connect to your email server.

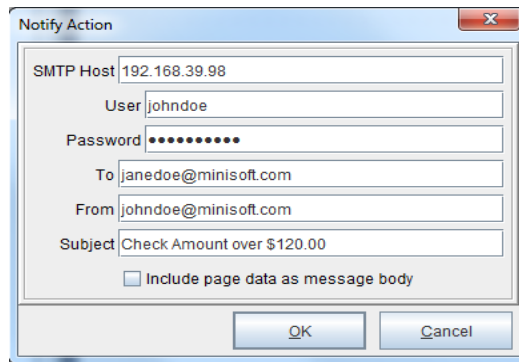
Password is your password to connect to the SMTP Host.

To is the intended recipient of the email.

From is the company email address.

Subject is the description of the specified email.

Include page data as message body includes the data page in the body of the email.



The image shows a 'Notify Action' dialog box with the following fields and values:

- SMTP Host: 192.168.39.98
- User: johndoe
- Password: [Masked]
- To: janedoe@minisoft.com
- From: johndoe@minisoft.com
- Subject: Check Amount over \$120.00
- Include page data as message body:

Once complete, click *OK*.

Output from Project

Output from Project is used to retrieve output from a separate project, that uses an independent data file, and include that output in the current project.

The screenshot shows the 'Output From Project Action' dialog box. It has a title bar with a close button (X). The dialog is divided into several sections:

- Position:** Contains three input fields: 'Horizontal position' (0.0), 'Vertical position' (0.0), and a 'Rotation' dropdown menu (0).
- Project:** A text field containing the path 'C:\Minisoft\FORMz_6\projects\Translated_2ST\sample...' with a browse button (...).
- Input Source:** Two radio buttons. The first is selected: 'Input is from a component of a composite data file'. Below it is a 'Component' text field. The second radio button is 'Input is the value of the rule variable'.
- Input Format and Character Set:** Two dropdown menus. 'Input format' is set to 'PCL' and 'Character set' is set to 'ROMAN8'.
- Parameters:** A large empty text area for parameters, with buttons for 'Add', 'Remove', 'Move up', and 'Move down' to its right.
- Page:** A dropdown menu set to 'Every page' and a small text field containing '1'.
- Repeat Interval Spacing:** A text field containing '0.0' with a note '*0=Use Form's lines per inch' below it.
- Conditional vertical increment:** An unchecked checkbox.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

eFORMz Composer Data Customization

An XML (Extensible Markup Language) file with a “.cdf” extension can also be used to specify all the separate data files for a project that uses this procedure:

```
<?xml version="1.0" encoding="UTF-8"?>
<CompositeData>
  <Component Name="PackingSlip">
    <InputData filename="projects/WMI/Collate/PackingSlip.scs"/>
  </Component>
  <Component Name="ShippingLabel">
    <InputData filename="projects/WMI/Collate/UPS.scs" />
  </Component>
  <Component Name="ShippingLabel1">
    <InputData filename="projects/WMI/Collate/Fedex1.scs" />
  </Component>
</CompositeData>
```

When using a composite data file (.cdf), the primary component is configured on the “Advanced” tab of the Project Properties.

Cross-Reference: For a use of the Output from Project Rule, follow along with the following tutorial:

<http://support.minisoft.com/index.php/creating-a-two-sided-label-w-a-reference-number-parts-i-and-ii/>

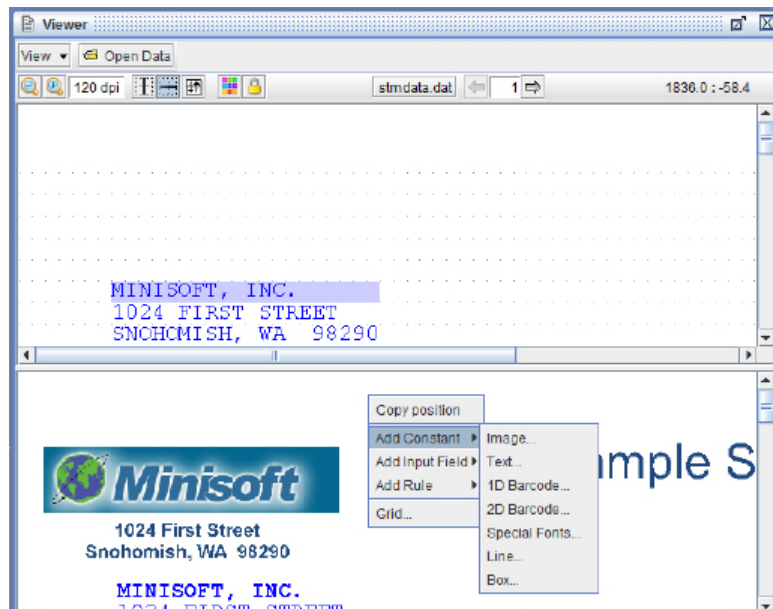
Constant Actions

Creating a Constant

eFORMz allows you to add constant values to your data output. Apply a constant value to your form that will display on every page. To create a constant value:

1. *Right click* in the eFORMz Form Viewer and select *Add Constant*.

TIP: If the position of the constant value needs to be adjusted simply drag the element into position using your mouse.



The options are as follows:

- ◆ Image, Text, 1D Barcode, 2D Barcode, Special Fonts, Line and Box

Image

Right click in the eFORMz Form Viewer and select *Add Constant > Image*. The Image Element dialog box will display:

In this example, a Minisoft logo from the project was loaded as a constant image. To load an image into a project, *right click* the project and select *Properties > Images*.

Options:

Name the image element with a descriptive value (optional).

Horizontal/Vertical Position sets the constant location (in decipoints).

Desired width and height are configured in decipoints (720 decipoints = 1 in).

Scaling options include *None*, *Shrink to desired size* and *Fit to desired size*.

None leaves the size as is. *Shrink to desired size* reduces the size of the image if it doesn't fit in the specified width and height. *Fit to desired size* will change the size of the image to match the specified width and height.

Cropping is enabled for images from a file. Images loaded from URLs or loaded directly into the project cannot be cropped.

SECTION 2

Default resolution sets the image resolution. The *Lock aspect ratio* option, when checked, will keep the ratio of the image's dimensions the same when scaling. It figures which ratio (desired width / image width or desired height / image height) is needed for scaling both dimensions of the image to make it shrink or fit within the desired rectangle. *Load from* options include:

Project

Select an image previous loaded into the project file through the *Project Properties > Image* dialog box.

File

Use the browse button to select the image you would like to add to your form as a constant value.

PDF File

Use the browse button to select a PDF file, then in the *Load Page* field, specify the page you would like to add to your form as a constant value.

Text

Right click in the eFORMz Form Viewer and select *Add Constant > Text*. The Text Element dialog box will display:

The screenshot shows the 'Text Element' dialog box with the following settings:

- Position:**
 - Name: VisitNote
 - Text: Note: Visit our website for further updates at www.minisoft.com
 - Horizontal position: 528.0
 - Vertical position: 174.0
 - Text Color: Black
- Underline:**
 - Style: None
 - Thickness: 7.2
 - Offset: 7.2
- Alignment:**
 - Rotation: 0
 - None (selected)
 - Left
 - Right
 - Center
 - Width: 0.0

Buttons at the bottom: Change font, OK, Cancel.

eFORMz Composer Data Customization

Enter the text string you would like to have added to your form as a constant value. Also, if needed, set the character alignment and font style. Once complete, click *OK*.

Options:

Name enter a descriptive name for the constant value (optional).

Text enter the text string.

Horizontal/Vertical Position the constant location in decipoints (180 decipoints = 1/4 inch).

Text Color adjusts the color of the text string (Color specifications: Swatches, HSV, HSL, RGB & CMYK).

Underline offers two styles: *single solid* and *double solid*. The thickness of the line and its offset from the value can be configured.

Alignment rotate text 90, 180 and 270 degrees (Default is 0). The location can be shifted left, right or center (Default is none).

Change Font adjusts the font style.

1D Barcode

Right click in the eFORMz Form Viewer and select *Add Constant > 1D Barcode*. The Barcode Element dialog box will display:

Enter the characters or numeric values you would like to barcode into the *Text* field. From the *Barcode type* drop down box select the barcode type you would like to use. Adjust the width, height, and alignment if needed. Once complete click *OK*.

Options:

Name enter a descriptive name for the constant value (optional).

Text enter the string.

Barcode Type

Types include: 3OF9, 3OF9EXT, 2OF5, CODABAR, CODE 11, CODE 128, EAN8, EAN13, JAN8, JAN13, MSI, UPCA, UPCE, POSTNET, USER-DEFINE and INTELLIGENT MAIL.

Generate Checksum when checked, generates a checksum.

Optimize barcode length optimizes the length associated with Code128 barcodes.

eFORMz Composer Data Customization

Trim Data when checked, trims the string.

Barcode Color adjusts the color of the barcode.

Horizontal/Vertical Position the constant location in decipoints (180 decipoints = 1/4 inch).

Width/Height default width is 10 and height is 300. Configured in decipoints (720 decipoints = 1 in).

Alignment rotate barcode 90, 180 and 270 degrees (Default is 0). The orientation can be left, right or center (Default is none).

Cross-Reference: For more information, see the 1D Barcodes part of the *Barcodes Mini-Manual*.

2D Barcode

Right click in the eFORMz Form Viewer and select *Add Constant > 2D Barcode*. The 2D Barcode Constant dialog box will display:

Enter the characters or numeric values you would like to barcode into the *Text* field. From the *Barcode type* drop down box select the barcode type you would like to use. Adjust the height and alignment if needed. Once complete click *OK*.

Options:

Name enter a descriptive name for the constant value (optional).

Text enter the string.

Barcode Type

Types include: MaxiCode, Aztec, Data Matrix, PDF₄₁₇ and Quick Response (QR).

Horizontal/Vertical Position the constant location in decipoints (180 decipoints = 1/4 inch).

Rotation rotate barcode 90, 180 and 270 degrees (Default is 0).

Cross-Reference: For more information, see the 2D Barcodes part of the *Barcodes Mini-Manual*.

Special Fonts

Right click in the eFORMz Form Viewer and select *Add Constant > Special Fonts*. The Special Fonts Element dialog box will display:

Options:

Name enter a descriptive name for the constant value (optional).

Text enter the string.

Barcode Type

Types include: MICR₁₃B, MICRCMC₇, OCRA and SECURE AMOUNT.

Horizontal/Vertical Position the constant location in decipoints (180 decipoints = 1/4 inch).

Special Fonts Color adjust the color of the barcode.

Alignment rotate barcode 90, 180 and 270 degrees (Default is 0). The location can be shifted left, right or center (Default is none).

Cross-Reference: For more information, see the Special Fonts part of the *Barcodes Mini-Manual*.

Line

Right click in the eFORMz Form Viewer and select *Add Constant > Line*. The Line Element dialog box will display:

Adjust the position, length and thickness as needed. If necessary, update the plane (horizontal or vertical) and/or the color (Default is black). Once complete, click *OK*.

Options:

Name

Enter a descriptive name for the constant value (optional).

Horizontal/Vertical Position

The constant location in decipoints (180 decipoints = 1/4 inch).

Length

Adjust the length of the line (in decipoints). Default is 720 (1 inch).

Thickness

Adjust the thickness of the line (in decipoints). Default is 10.

Line Color

Adjust the color of the line.

Horizontal/Vertical

The orientation of the line.

Box

Right click in the eFORMz Form Viewer and select *Add Constant > Box*. The Box Element dialog box will display:

The screenshot shows the 'Line Element' dialog box with the following fields and values:

- Name: Box
- Horizontal position: 1986.0
- Width: 720.0
- Vertical position: 395.0
- Height: 720.0
- Border thickness: 10.0
- Border Color: Black
- Fill: (unchecked)
- Fill Color: White

Adjust the position, width and height as needed. If necessary, update the border thickness and/or the border color (Default is black). To add a fill color, check the *Fill* box. Update the fill color as seen fit (Default is white). Once complete, click *OK*.

Options:

Name enter a descriptive name for the constant value (optional).

Horizontal/Vertical Position the constant location in decipoints (180 decipoints = 1/4 inch).

Width adjust the width of the box (in decipoints). Default is 720 (1 inch).

Height adjust the height of the box (in decipoints). Default is 720 (1 inch).

Border Thickness adjust the thickness of the border.

Border Color adjust the color of the border.

Fill when checked, fills the box with a specified color (default unchecked).

Fill Color adjust the fill color.

Conditional Processing

eFORMz allows you to place conditions on rules and procedures. Setting a condition on a rule/procedure determines whether or not the action associated with the rule/procedure is processed.

eFORMz supports these logical operators:

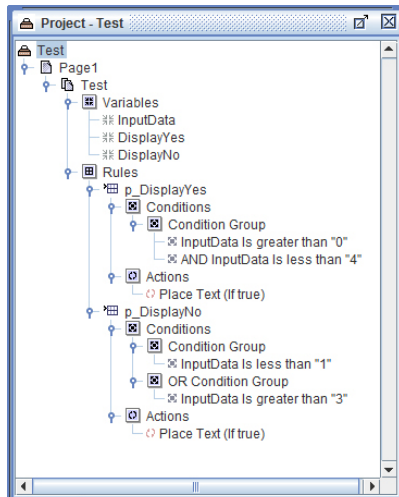
- ◆ AND
- ◆ OR
- ◆ NOT

And the following data conditional operators:

Is equal to	Starts with	Is a number
Is not equal to	Ends with	Names a file that exists
Is less than	Contains	
Is greater than	Matches	

How to implement

eFORMz allows you to place conditions on Rules and Procedures. In the below example the word "YES" is displayed (variable DisplayYes) if the value of the variable InputData is equal to 1, 2, or 3. The word "NO" is displayed if the value if InputData is not 1, 2, or 3.



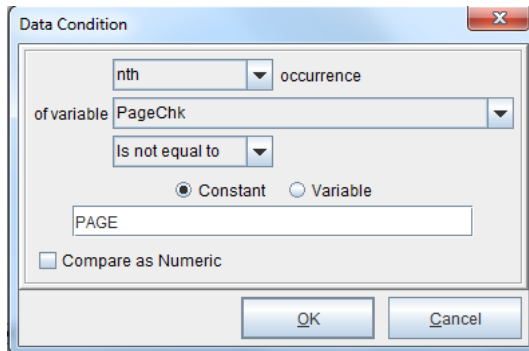
Adding a Condition

The following example will demonstrate how a project can be implemented for conditional logic. Let's say, there is a need to add special wording to an output file based on the state (Colorado or Oklahoma) and if it is the last page of the order:

1. Create two variables, one looking at your state check and another looking for a check if it is the final page of the order. In this example we will be using the following variables:

PageChk and State_Chk

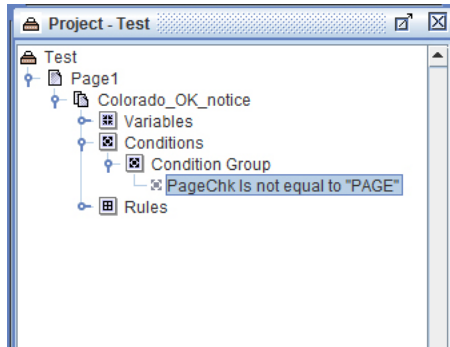
2. Right click your form and select *Add AND Condition*, followed by selecting the data conditional operator you would like to work with. For this example, the conditional operator used for our first variable will be '*Is not equal to*'. Once selected the *Data Condition* dialog box displays.
3. Select the variable you would like the condition to be based on (*PageChk*) and then confirm your conditional operator along with your comparison value. For this example, the comparison value will be on the word '*PAGE*':



Once complete, select *OK*.

SECTION 2

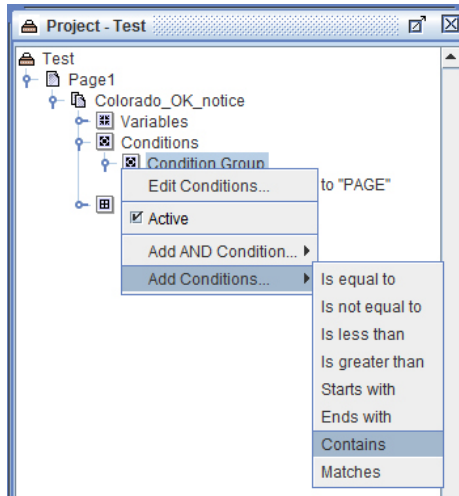
4. Your new condition will now appear under *Condition > Condition Group*:



Next, add the second conditional check (*State_Chk*). Since the second conditional check has multiple options (Colorado OR Oklahoma) the condition becomes an 'AND Condition Group'.

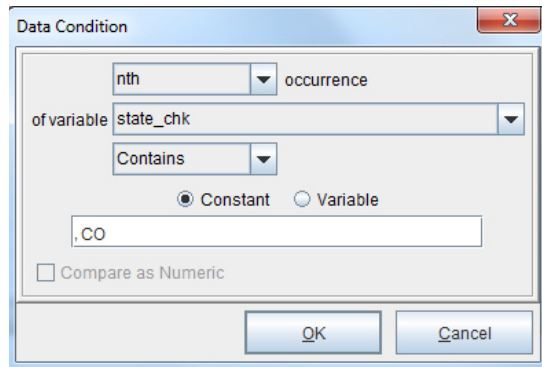
To create an 'AND Condition Group' containing OR conditions:

1. Right click *Condition Group* and select *Add Conditions*. Select a conditional type, for example *Contains*:

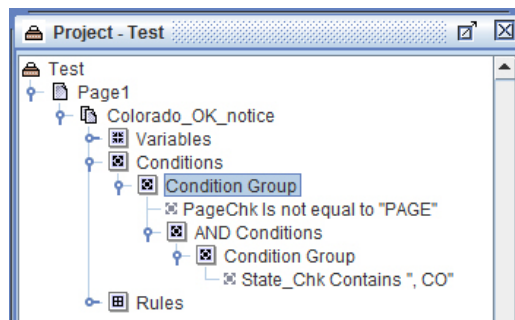


eFORMz Composer Data Customization

2. The Data Condition dialog box displays. Select the variable you would like the condition to be based on (*State_Chk*) and then confirm your conditional operator along with your comparison value. For this example, the comparison will be on the state of Colorado (*CO*):



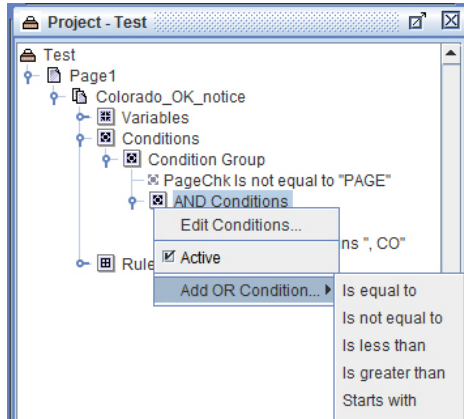
3. Your new condition will now appear under *AND Condition Group*, as show below:



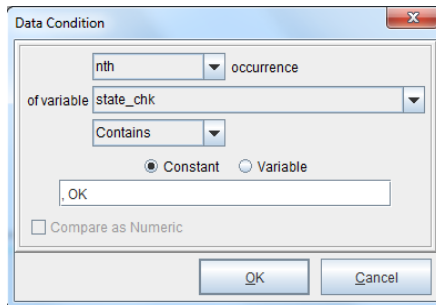
4. The next step is to create the second conditional check, causing the AND Condition Group to contain an OR option since we are looking for multiple comparisons (Colorado *OR* Oklahoma).

SECTION 2

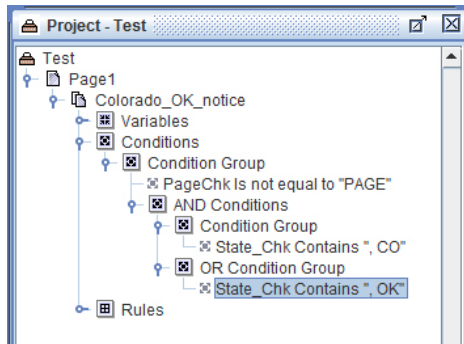
5. Right click '*AND Condition Group*' and select *Add OR Condition*. Select a conditional type, for example *Contains*:



6. The Data Condition dialog box displays. Select the variable you would like the condition to be based on (*State_Chk*) and then confirm your conditional operator along with your comparison value. For this example, the comparison will be on the state of Oklahoma (*OK*):



7. An *OR* condition will then appear with your '*AND Condition Group*', as shown below:



Appendix A

Advanced Processing

Command File Processor

Script Example

```
java -jar Run.jar - com.minisoft.HostTools.Local.eFORMz -d c:\Minisoft\eFORMz_6\input -f
*.dat:rename -p project.efz -t PrinterA
```

Parameters

In this example, the java executable is assumed to be in the path of the dos shell. The printer name is PrinterA as defined in the Printers.xml file. The project file (project.efz) is in the current directory.

-v	Log to stdout
-l <filename>	Log to new <filename>
-a <filename>	Append log to <filename>
-g <number>	Log priority fence
-d <directory>	Specify input directory
-f <filename[:disposition]>	Specify file selection and optionally disposition
-p <project>	Specify project
-t <printer>	Print (Black and White PCL)
-tc <printer>	Print (Color PCL)
-tp <printer>	Print (Postscript)
-o <filename>	Output to file
-u <basefilename>	Output to file with unique name
-e <basedirectory>	Process with eDIRECT
-x <command>	Execute external command after output
-c <directory>;<filemask>;<copy to directory>	Copy files after output
-r <directory>;<filemask>	Delete files after output
-n <directory>;<filemask>;<printer>	Print files after output
-help	

Preprocessor Data Parsing Control Codes

<code>\b</code>	backspace
<code>\t</code>	tab
<code>\n</code>	linefeed
<code>\r</code>	carriage return
<code>\f</code>	formfeed
<code>\\</code>	backward slash
<code>\a<3 decimal digits></code>	character represented by its decimal value (000-255)
<code>\x<2 hexadecimal digits></code>	character represented by its hexadecimal value (00 - ff)

eSUBMIT Command Processing

Overview

This utility of eFORMz permits host-independent redirection of application data. Originally designed to simply write data destined for standard output device to unique file names in a local directory for processing by the eFORMz local queue monitor.

While still accomplishing the original goal, eSubmit has been expanded to handle a variety of file redirection tasks independent of host operating system or third party subsystems.

Details

For parameters with characters that may be handled by the host command processor (such as semicolons on UNIX, use double quotes around the parameter value:

```
java -jar jars/eSubmit.jar -lpr "FILEPRINT;TITLEA;lphost"
```

Commands

File Name (-f)

Specify file selection and optionally disposition. Optional dispositions include: rename and delete

Example: -f "inv.data:rename"

Output Directory (-d)

Specify input directory.

Example: -d "/minisoft/salesorders"

MQ Queue (-mq)

This option directs the data to an IBM® MQ Queue on a remote system.

Message Queue Format (-mqm)

Specify message format.

Secure FTP (-rd)

This option has five parameters; remote directory, host, port, user name, password.

Use the File Name (-f) option to set the file name mask.

Project (-p)

Specify project file.

Example: -p "sampinv.efz"

LPR (-lpr)

Send data to an LPD service. Three required parameters.

The first parameter is the queue name on the remote host.

The second parameter is used in the naming of the file on the receiving host. Use it to direct the queue monitor in what project to use. The title can also be used in defining which output device eFORMz will use.

The fourth parameter, used to define the port to connect to, is optional.

Secure LPR (-slp)

Send data securely to an LPD service. Eight required parameters.

The first parameter is the queue name on the remote host.

The second parameter is used in the naming of the file on the receiving host. Use it to direct the queue monitor in what project to use. The title can also be used in defining which output device eFORMz will use.

The fourth parameter, used to define the port to connect.

PCL Print, Black and White (-t)

PCL Print, Color (-tc)

Printer name.

Example: -t "LJ5si" or -tc "LJ3800DN"

Output to File (-o)

Output to file. Output file extension options include: .pdf, .prn, .pcl, .xml.

Example: -o "inv.pdf"

APPENDIX A

Output to unique file name (-u)

Output to file with unique names. Output file extension options include: .pdf, .prn, .pcl, .xml.

Example: `-u "inv_.pdf"`

Process with eDirect (-e)

Process with eDirect. The base directory will be the directory the eDirect output will be sent to.

Example: `-e "/minisoft/invoice/output"`

Process with eDirectPlus (-ep)

Process with eDirectPlus. The base directory will be the directory the eDirectPlus output will be sent to.

Example: `-ep "/minisoft/invoice/output"`

Execute external command after output (-x)

Execute external command after output. Command options: mv - move command

Example: `-x "mv /minisoft/input/*.hld /minisoft/backup"`

Copy files after output (-c)

Copy files after output.

Example: `-c "/minisoft/input/*.hld /minisoft/backup"`

Delete files after output (-r)

Delete files after output.

Example: `-r "/minisoft/input/*.hld"`

Print files after output (-n)

Print files after output.

Example: `-n "/minisoft/output/*.prn roadrunner"`

Static input file (-i)

This option is used to read from a fixed file in place of the standard input device. It is useful to simulate input during testing.

Log file location and name (-l)

This option has two required parameters, the directory in which to create log files and the file mask used to create log files.

Example: `-l "/minisoft/log;iformzlog.txt"`

Verbose tracing (-v)

Use this option at the direction of Minisoft support to identify configuration issues.

Timeout in seconds (-w)

While reading stdin (or a static file) time to wait for data or EOF.

Execute after data is copied (-exe)

It will launch a local queue monitor to execute against the copied file using the configuration set by the other arguments or configuration file.

So if you used the arguments:

`-d <directory> -f<file> -p<project> -t <printer>`

eSubmit would copy the stdin to a unique file in <directory> and execute a queue monitor to execute a project against that file to print to <printer>

APPENDIX A

If you used the arguments:

<iqname> -exe

eSubmit would use the settings in the configuration file for <iqname> to copy the file and execute a queue monitor to process the file using those same settings, except the file selection will be set to the destination file name.

eFORMz is needed with a valid license.

Configuration (-cfg)

Use configuration file. Defaults to eSubmit.cfg. The configuration is based upon the monitor object and read looking for a matching InputQueue. The queue name to match is the parameter if only one bare (without a leading hyphen) parameter is supplied. If there is not a single bare parameter, an attempt will be made to locate a blank (no characters) queue name.

Help (-help)

Display utility parameter usage.

Quick Reference Guide

-f <filemask[:disposition]>	Specify file name
-d <directory>	Specify output directory
-mq <host>;<port>;<channel>;<queue manager>;<message queue>	Specify MQ queue
-mqm <file format>	Specify message format
-rd <remote directory>;<host>;<port>;<user>;<password>	Specify remote (Sftp) output directory
-p <project>	Specify project
-lpr <queue>;<title>;<host>[:<port>]	Send data to an LPD service

-slp <queue>;<title>;<lpd host>;<lpd port>;<ssh host>;<sshport>;<user>;<password>	Send data to an LPD Service via SSL
-t <printer>	Print (Black and White PCL)
-tc <printer>	Print (Color PCL)
-tp <printer>	Print (Postscript)
-o <filename>	Output to file
-u <basefilename>	Output to file with unique name
-e <basedirectory>	Process with eDIRECT
-ep <basedirectory>	Process with eDirectPlus
-x <command>	Execute external command after output

APPENDIX A

-c <directory>;<filemask>;<directory>	Copy files after output
-r <directory>;<filemask>	Delete files after output
-n <directory>;<filemask>;<printer>	Print files after output
-i <filename>	Specify static input file (in place of stdin)
-l <directory:filemask>	Specify logfile location and name
-v	Verbose (stdout) tracing
-w <number>	Time in seconds to wait for more data (default 30)
-exe	Execute after the data is copied
-cfg	Configuration filename (defaults to eSubmit.cfg)
-help	

User Configuration Files

Site specific configuration files available:

eFORMzCfg_User.xml
FontSpecs_User.xml

The above files are not installed with an eFORMz installation to insure that the files are never overwritten. Files with the above naming convention should be placed in the following directory:

C:\Minisoft\eFORMz_6\

Custom Form Sizes

Site specific custom form sizes can be added to the *eFORMzCfg_User.xml*, for example:

```
<PageDimension Name="B6" PCLNumber="1" PortraitA="5250" PortraitB="7560" PortraitC="4860"
PortraitD="7560" PortraitE="180" PortraitF="0" PortraitG="120" PortraitH="360" LandscapeA="7560"
LandscapeB="5250" LandscapeC="7272" LandscapeD="5250" LandscapeE="144" LandScapeF="0"
LandscapeG="120" LandscapeH="360" />
```

Contact Minisoft (support@minisoft.com) for custom form sizes.

Cross-Reference: For more information on adding a custom form or label size, view the following tutorial:

<http://support.minisoft.com/index.php/custom-form-and-label-sizes/>

Preprocessors and External Functions

Place preprocessors and/or external function references in the following file:

eFORMzCfg_User.xml

Minisoft will provide a user with the custom file reference and external files needed, for example:

```
<?xml version="1.0" encoding="UTF-8"?>
<eFORMzConfiguration>
<Preprocessor Name="Macro Insert" Class="com.sample.MacroInsertPreprocessor" />
<ExternalFunction Name="eVS Barcode Data Generator"
Class="com.minisoft.ExternalFunctions.eVSBarcodeDataGenerator" />
</eFORMzConfiguration>
```

Font Configuration

Site specific font configurations can be placed in a file referred as follows:

FontSpecs_User.xml

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<Fonts>
  <Font Name="Arial Black" PCLTypeface="-1" Spacing="Proportional"
    PitchFactor="0.008332330903266" ViewerFont="Arial Black" PDFFont="fonts/ariblk.ttf">
    <Style Style="4" Factor=".66667" />
    <Style Style="5" Factor=".66667" BaseStyle="1" />
    <Style Style="8" Factor=".5" />
    <Style Style="24" Factor="1.5" />
  </Font>
<RTFInputFonts>
  <Font InputName="Arial Black" FontName="Arial Black" />
</RTFInputFonts>
</Fonts>
```

Cross-Reference: For more information on adding a font to eFORMz, view the following tutorial:

<http://support.minisoft.com/index.php/adding-a-font-to-eformz/>

Appendix B

Database URL and Driver Classes

Supported URL and Driver Classes

DB2 (iSeries)

Database name:

Connection name. Actual database name not needed in this field.

Driver Class:

com.ibm.as400.access.AS400JDBCdriver

URL:

jdbc:as400://123.456.78.90

Add the following properties and their values:

user

password

databasename

If you do not know the database name, logon to your AS/400 and run WRKRDBDIRE from the command line. Only the database marked *LOCAL is the local database that resides on the AS/400 system.

Image and Eloquence

Database name:

Connection name. Actual database name not needed in this field.

Driver Class:

com.minisoft.jdbc.MSJDBCdriver

URL:

Write access:

jdbc:MSJDBC:///?Server=xxx.xxx.xx.xxx&ServerPort=30007&User=xxx&UserPassword=xxx&Group=xxx&Account=xxx&ImageDatabaseo=eformz,WRITE,0,1,0

Read access:

jdbc:MSJDBC:///?Server=xxx.xxx.xx.xxx&ServerPort=30007&User=xxx&UserPassword=xxx&Group=xxx&Account=xxx&ImageDatabaseo=eformz,READALL,0,5,0

Note: Requires Minisoft's JDBC (Java Database Connectivity) driver.

Informix

Driver Class:

```
com.informix.jdbc.IfxDriver
```

URL:

```
jdbc:informix-sqli://<IP address of host machine>:<port no>/<database>:  
INFORMIXSERVER=<server_name>;user=<user_name>;password=<password>
```

Example:

```
jdbc:informix-sqli://Sturgeon:9088/stores_demo:  
INFORMIXSERVER=informix_server;user=admin;password=xxx
```

Microsoft Access

Driver Class:

```
sun.jdbc.odbc.JdbcOdbcDriver
```

URL direct path example:

```
jdbc:odbc:DRIVER={Microsoft Access Driver (*.mdb)};  
DBQ=c:\\Minisoft\\eFORMz_6\\Projects\\eDirect\\Sample3\\emaildb.mdb
```

Note: Path specification is case sensitive.

URL using system DSN:

```
jdbc:odbc:[database name]
```

Example:

```
jdbc:odbc:emaildb
```

Create a System DSN

1. From your Start menu, select *Settings > Control Panel*.
2. The Control Panel dialog box appears. Select *Administrative Tools* and *ODBC Data Source*.
3. The ODBC Data Source Administrator window appears. Select the *System DSN* tab and then click *Add*.

APPENDIX B

4. The Create New Data Source dialog box appears. Select *Microsoft Access Driver (*.mdb)* from the list of drivers. Click *Finish*.
5. The ODBC Microsoft Access Setup dialog box appears.
6. From the Database section click on *Select* to establish the path to your database, located on your PC.
7. Once the Database path has been established, enter a database reference name in the Data Source Name field.
8. Enter a database description, if needed, in the Description field. Once complete click *OK*.
10. Your newly created database reference will appear in the ODBC Data Source Administrator dialog box under the System DSN tab. Click *OK*.

MySQL

Driver Class:

`com.mysql.jdbc.Driver`

URL:

`jdbc:mysql://[host name or ip address]/[database name]?user=[user name]&password=[user password]`

`jdbc:mysql://123.456.78.90/eformz?user=eformz&password=eformz`

Note: When referencing the database url address in the web.xml file, the user name and password values can be excluded. For example:

`jdbc:mysql://123.456.78.90/eformz`

Note: In order for eFORMz to connect to your MySQL database you will need to copy your MySQL jdbc driver class into your eFORMz jar folder:

Copy:

`\sqlDriver\mysql-connector-java-xxx\mysql-connector-java-xxxx-bin.jar`

to the following directory:

`C:\Minisoft\eFORMz_6\jars\Local`

Oracle

Driver Class:

`oracle.jdbc.driver.OracleDriver`

URL:

`jdbc:oracle:thin:[user]/[password]@[hostname or ip address]:[Listener Port]:
[SID – Database Name]`

Example:

`jdbc:oracle:thin:@123.456.78.90:1521:emaildb`

Note: In order for eFORMz to connect to your Oracle database, you will need to copy your oracle jdbc driver class into your eFORMz jar folder:

Copy:

`[Oracle_Home]\jdbc\lib\classesXXX.zip`

to the following directory:

`C:\Minisoft\eFORMz_6\jars\Local`

and then rename the classesXXX.zip to classesXXX.jar

Oracle Cluster

JDBC URL for Oracle RAC (Real Application Cluster)

`jdbc:oracle:thin:[user]/[password]@(DESCRIPTION=(LOAD_BALANCE=on)
(ADDRESS=(PROTOCOL=TCP)(HOST=host1) (PORT=1521))
(ADDRESS=(PROTOCOL=TCP)(HOST=host2) (PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=service)))`

APPENDIX B

Progress

Driver Class:

`com.ddtek.jdbc.openedge.OpenEdgeDriver`

URL:

`jdbc:datadirect:openedge://servername:10005;databaseName=c:\progressdb\ sportstest;user=username;password=password;defaultschema=pub`

JAR Files:

From your *Progress* installation folder copy the following files into the C:\Minisoft\ eFORMz_6\jars\Local\ folder:

`base.jar`
`openedge.jar`
`util.jar`

SQL Server

Driver Class:

`com.microsoft.sqlserver.jdbc.SQLServerDriver`

URL:

`jdbc:sqlserver://12.34.567.899:1433;databaseName=ECOMVER;
user=minisoft;password=minisoft2007;selectMethod=cursor;
parametersAsUnicode=false`

Note, you will need to specify the correct IP address, port (if different), database name, user, and password.

Sybase

Driver Class:

```
com.sybase.jdbc3.jdbc.SybDriver
```

URL:

```
jdbc:sybase:Tds:localhost:2638
```

Parameters:

```
SERVICENAME=demo  
HOSTNAME=localhost  
user=DBA  
password=sql  
STREAM_CACHE_SIZE=10000  
LITERAL_PARAMS=false  
CURSOR_ROWS=0  
SESSION_TIMEOUT=300  
JCONNECT_VERSION=6  
CANCEL_ALL=true  
DYNAMIC_PREPARE=true  
LANGUAGE_CURSOR=false  
SEND_LONG_PARAMS REGARDLESS_OF_CAPABILITIES=false  
IGNORE_DONE_IN_PROC=false  
REQUEST_HA_SESSION=false  
ELIMINATE_o1oSM=true  
GET_BY_NAME_USES_COLUMN_LABEL=false  
ESCAPE_PROCESSING_DEFAULT=false
```

JAR File:

Place the Sybase jar file into the C:\Minisoft\eFORMz_6\Local\ folder:

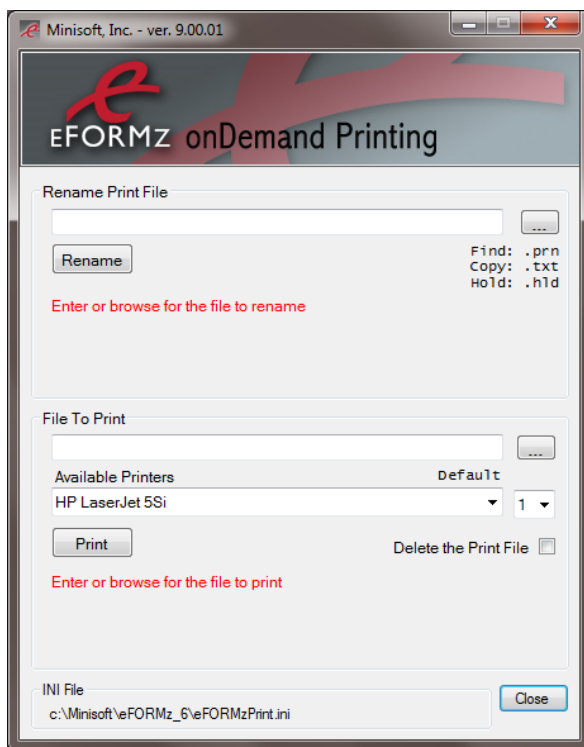
```
jconn3.jar
```

Appendix C

OnDemand Print Utility

OnDemand Print

The OnDemand Print utility, which is included with the install, allows one to enter data as a value via keyboard entry and/or a scanner. It also includes an option to use a browse button to select file(s) from specific folders. There are two sections within this utility. One is to Rename the extension of a (print) file, which is often used for the later capture and processing of the a print file by the eFORMz print monitor. The other is to simply select and Print a file.



An .ini file is included which allows one to configure the directory path, a specific 'find' extension, a 'renamed' extension and an already processed extension (.hld) which matches your needs.

APPENDIX C

For example, if a match is found this utility will rename the present .prn extension to .rdy or .dat. The eFORMz Local Toolkit can then be modified to look for .rdy /.dat files. Once the eFORMz monitor selects and processes these files it will automatically change the extension back to .hld (hold).

eFORMzPrint.ini

[eFORMzPrint Options]

;DirectoryPath: Path to the print files to be renamed
;FindExtension: File extension of the file to be renamed
;CopyExtension: File extension of the renamed file
;HoldExtension: File extension of already processed print files
;BrowseOnlyFind: 0=false 1=true. When set to 0 all files are displayed when browsing
; and hold files can be reprinted.
; When set to 1 only files with the find extension are
; displayed when browsing and only files with the find
; extension will be printed - no reprinting of hold files.

DirectoryPath=c:\
FindExtension=.prn
CopyExtension = .rdy
HoldExtension=.hld
BrowseOnlyFind=0

Index

Index

A

- advanced options 26
- archives option 30
- auto-save project 6

C

- character height
 - form/data viewer 13
- characters per inch
 - form/data viewer 13
- command parsing 27
- conditional processing 162
- constant values
 - 1D barcode 156
 - 2D barcode 158
 - box 161
 - image 153
 - line 160
 - special fonts 159
 - text 154

D

- database
 - driver classes 180
 - url 180
- database configuration
 - project option 29
- data color
 - form/data viewer 7
- data file preprocessor 27
- data grid 15
- data parsing options 24
- DB2 (iSeries)
 - database configuration 180
- driver classes 180

E

- eFORMzCfg_User.xml 176
- eFORMz system properties 6
- Eloquence
 - database configuration 180
- eSUBMIT command processing 169
- eVS barcode 100

F

- file menu
 - new project 2
 - open project 2
 - save project 2
 - save project as 2
 - system properties 2
- font configuration 178
- FontSpecs_User.xml 176
- form and data viewer
 - characters per inch 13
 - data color 14
 - lines per inch 13
 - options 13
- form properties
 - bold 42
 - character set 41
 - chars per inch 41
 - data viewer size 40
 - exclude data 40
 - font 41
 - italic 42
 - lines per inch 40
 - output tray selection 41
 - overlay on previous form 40
 - page orientation 39
 - paper side 40
 - paper size 39
 - source tray 40
 - text color 42
 - top and left offsets 40
- form sizes
 - custom 177
- functions
 - base64 decode 105

- base64 encode 105
- bulk concatenate 104
- choose one of 83
- concatenate 63
- does file exist 99
- eVS barcode data generator 100
- hashsha1 107
- insert string 67
 - attach string to existing data 70
- lowercase and uppercase 90
- modulo 107
- parse date 87
- regex replace 106
- replace 79
- right and left trim 88
- set value 72
 - setting an image value 76
- set value from a declared constant 84
- set value from a keyed declared constant 85
- SmartLabel design and barcode 102
- SQL lookup 97
- string length 108
- string position 108
- substring 77
- table lookup 95
- translate 86
- UPC-EAN check digit 94
- URL decoder 93
- URL encoder 92

H

- help menu
 - about Minisoft eFORMz 5
- host tools menu
 - AS400 4
 - local toolkit 4
 - unix 4
- HTML template 47

I

- Image
 - database configuration 180
- import image 28
- Informix
 - database configuration 181

L

- lines per inch
 - form/data viewer 13
- line terminator 26
- load image 140
- load initial page number from a file 27

M

- menu options
 - file menu 2
 - project menu 2
 - run menu 3
- message priority fence 7, 11
- Microsoft Access
 - database configuration 181
- minimum page number size 26
- MySQL
 - database configuration 182

O

- Oracle
 - database configuration 183
- output tray selection 41

P

- page properties 37
 - xml input 38
- pdf output option 32
- place text 132
 - rotating text 136
 - wordwrap 137
- preprocessor
 - character set translator 28

- command file processor 168
- insert a string 28
- macro insert 27
- multiple replace occurrences of a string 27
- replace occurrences of a string 27
- sequential line marker 27, 28
- skip until a string is found 27
- text file pagination 27
- printer controls
 - base macro 20
 - copies 20
 - exclude unprintable edge in PDF output 20
 - left offset registration 20
 - output to xml verbose mode 20
 - perforation skip 20
 - print/output all pages 21
 - print/output current pages 21
 - print/output range of pages 21
 - simplex/duplex 20
 - top offset registration 20
- procedure
 - calculation 110
 - call a dynamic service 128
 - call a web service 127
 - date calculation 113
 - date format 116
 - file list 129
 - load a component 126
 - load variable from a declared constant 122
 - load variable from a keyed declared constant 123
 - load variable from a variable 124
 - load variable from project output 124
 - load variable from xml node 125
 - numeric format 115
 - printer list 130
 - set variable 112
 - sort 114
 - SQL end transaction 121
 - SQL load 118
 - sql update 119
 - SQL update 118

- sum 113
- time interval format 117
- Progress
 - database configuration 184
- project menu
 - add form to page... 3
 - add new page to project... 2
 - properties 3
 - reload form 3
 - remove selected item 3
 - replace form 3
- project notes 23
- project properties 19

R

- replace form 3
- replace occurrences of a string 27
- Right and Left Trim 88
- rotating text 136
- rule
 - actions
 - barcode 131
 - charts 131
 - load image 140
 - notify 145
 - output from project 150, 151
 - page number 145
 - place text 132
 - translate 146
 - load image
 - load image based on a condition 143
- run menu
 - output.. 3
 - print 3
 - process w/e-Direct 4

S

- scrollbars 8, 14
- sequential line marker 27
- skip string 27

- SmartLabel
 - creating 102
 - sorting 21
 - source tray option 40
- SQL Server
 - database configuration 184
- Sybase
 - database configuration 185
- system properties 6
 - acrobat reader 9
 - auto-load viewer 8
 - automatically view eFORMz document output 9
 - automatically view pdf output 9
 - auto-save project 6
 - compress pdf output 9
 - constant color 7
 - data color 7
 - default data path 6
 - default output path 9
 - default project path 6
 - embed fonts in pdf output 9
 - environment 6, 7
 - horizontal split pane 8
 - log file 10
 - log to file 10
 - log to standard output 10
 - message priority fence 7, 11
 - output to XML in verbose mode 9
 - roll count 10
 - show offset toolbars 8
 - synchronized scrollbars 8
 - variable color 8
 - vertical split pane 8
 - viewer resolution 7
- system variables 59

T

- text to XML converter 27

U

- user configuration files 176

V

- variable
 - form variables 50
 - by position 50
 - for activeForms input 54
 - queue monitor variable 55
 - using data node 52
 - global variables 56
 - options
 - bookmark in PDF output 51
 - copy original data to output 51
 - height 51
 - horizontal/vertical position 51
 - include variable in XML output 51
 - increment line by 51
 - keep spacing 51
 - name 51
 - # of characters 51
 - repeat 51
- viewer controls
 - color control 12
 - previous/next buttons 12
 - row and column coordinates 12
 - vertical/horizontal panes 12
 - zoom control 12
- viewer options
 - characters per inch 13
 - constant color 14
 - data color 14
 - data grid
 - show grid 15, 16
 - snap selection to grid 15
 - height 13
 - lines per inch 13
 - offset adjustment toolbars 13
 - synchronize scrollbars 14
 - variable color 14
 - variables viewer 17

W

window menu
 eFORMz Composer 5
wordwrap 137

X

xml output option 31

Z

zebra xml 46
zoom control 12