

# STYLUS STUDIO 2007

XML DEPLOYMENT COMPONENTS  
ADAPTER URL REFERENCE





## Stylus Studio® 2007 XML Deployment Components Adapter URL Reference

Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. This manual is also copyrighted and all rights are reserved. This manual may not, in whole or in part, be copied, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from Progress Software Corporation.

The information in this manual is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear in this document. The references in this manual to specific platforms supported are subject to change.

A (and design), Actional, Actional (and design), Affinities Server, Allegrix, Allegrix (and design), Apama, Business Empowerment, DataDirect (and design), DataDirect Connect, DataDirect Connect64, DataDirect Connect OLE DB, DataDirect Technologies, DirectAlert, Dynamic Index Utility, Dynamic Routing Architecture, EasyAsk, EdgeXtend, Empowerment Center, eXcelon, Fathom, Halo, IntelliStream, Iwave Integrator, Neon Unload, O (and design), ObjectStore, OpenEdge, PDF, PeerDirect, Persistence, Persistence (and design), POSSENET, Powered by Progress, PowerTier, ProCare, Progress, Progress Dynamics, Progress Business Empowerment, Progress Empowerment Center, Progress Empowerment Program, Progress Fast Track, Progress OpenEdge, Progress Profiles, Progress Results, Progress Software Developers Network, ProtoSpeed, ProVision, SequeLink, Shadow, ShadowDirect, Shadow Interface, Shadow Web Interface, Shadow Web Server, Shadow TLS, Sonic ESB, SonicMQ, SOAPStation, Sonic Software (and design), SonicSynergy, Speed Load, SpeedScript, Speed Unload, Stylus Studio, Technical Empowerment, Visual Edge, Voice of Experience, WebSpeed, and Your Software, Our Technology-Experience the Connection are registered trademarks of Progress Software Corporation or one of its subsidiaries or affiliates in the U.S. and/or other countries. AccelEvent, A Data Center of Your Very Own, Apama Dashboard Studio, Apama Event Manager, Apama Event Modeler, Apama Event Store, AppsAlive, AppServer, ASPen, ASP-in-a-Box, BusinessEdge, Cache-Forward, Connect Everything, Achieve Anything., DataDirect Spy, DataDirect SupportLink, DataDirect XQuery, DataXtend, Future Proof, Ghost Agents, GVAC, Looking Glass, ObjectCache, ObjectStore Inspector, ObjectStore Performance Expert, Pantero, POSSE, ProDataSet, Progress DataXtend, Progress ESP Event Manager, Progress ESP Event Modeler, Progress Event Engine, Progress RFID, PSE Pro, PS Select, SectorAlliance, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, Sonic Business Integration Suite, Sonic Process Manager, Sonic Collaboration Server, Sonic Continuous Availability Architecture, Sonic Database Service, Sonic Workbench, Sonic Orchestration Server, Sonic XML Server, WebClient, and Who Makes Progress are trademarks or service marks of Progress Software Corporation or one of its subsidiaries or affiliates in the U.S. and other countries.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Any other trademarks or service marks contained herein are the property of their respective owners.

### Acknowledgments for Stylus Studio

Stylus Studio includes software developed by the Apache Software Foundation (<http://www.apache.org/>). Copyright © 1999-2000 the Apache Software Foundation. All rights reserved. The names "Xalan", "FOP", and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. For written permission, please contact [apache@apache.org](mailto:apache@apache.org).

Stylus Studio includes files that are subject to the DSTC Public License (DPL) Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.dstc.com>. Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License. The Original Code is xs3p. The Initial Developer of the Original Code is DSTC. Portions created by DSTC are Copyright © 2002. All rights reserved.

Stylus Studio includes software developed by DecisionSoft Limited. Copyright © 2001 DecisionSoft Limited. All rights reserved.

Stylus Studio includes software developed by Thai Open Source Software Center Ltd. Copyright © 2001-2003, Thai Open Source Software Center Ltd. All rights reserved.

Stylus Studio includes software developed by IBM. Copyright © 1995-2003 International Business Machines

Corporation and others. All rights reserved.

Stylus Studio includes JavaMail and JavaBeans Activation Framework software developed by Sun Microsystems. Copyright © 1994-2006, Sun Microsystems, Inc. All rights reserved.

Stylus Studio includes software developed by Kevin Atkinson. Copyright © 2000-2004, by Kevin Atkinson. All rights reserved.

Stylus Studio includes software developed by Maxim Shemanarev (McSeem) and is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. Copyright © 2002-2005 Maxim Shemanarev (McSeem). All rights reserved.

Stylus Studio includes Aspell 0.60.2, from the Free Software Foundation, Inc. (<http://www.fsf.org/>), which is subject to the GNU Lesser General Public License Version 2.1 (<http://www.gnu.org/licenses/lgpl.html>). Software distributed under this license is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the license for the specific language governing rights and limitations under the license.

Stylus Studio includes software developed by xqDoc.org. Copyright © 2005 Elsevier, Inc. All rights reserved.

Stylus Studio includes software developed by the World Wide Web Consortium. Copyright © 1994-2002 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/Consortium/Legal/>. All rights reserved.

Stylus Studio includes software developed by Info-ZIP. Copyright © 1990-2004 Info-ZIP. All rights reserved. For the purposes of this copyright and license, "Info-ZIP" is defined as the following set of individuals: Mark Adler, John Bush, Karl Davis, Harald Denker, Jean-Michel Dubois, Jean-loup Gailly, Hunter Goatley, Ian Gorman, Chris Herborth, Dirk Haase, Greg Hartwig, Robert Heath, Jonathan Hudson, Paul Kienitz, David Kirschbaum, Johnny Lee, Onno van der Linden, Igor Mandrichenko, Steve P. Miller, Sergio Monesi, Keith Owens, George Petrov, Greg Roelofs, Kai Uwe Rommel, Steve Salisbury, Dave Smith, Christian Spieler, Antoine Verheijen, Paul von Behren, Rich Wales, Mike White. Info-ZIP software is provided "as is", without warranty of any kind, express or implied. In no event shall Info-ZIP or its contributors be held liable for any direct, indirect, incidental, special or consequential damages arising out of the use of or inability to use this software.

Stylus Studio includes software developed by Mark Logic Corporation. Mark Logic, Mark Logic XDBC Java API and the Mark Logic logos are the trademarks of Mark Logic Corporation. Copyright © 2002-2005 Mark Logic Corporation. All rights reserved.

Stylus Studio includes software developed by Tim Bray and Sun Microsystems and is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. Copyright © 2004 Tim Bray and Sun Microsystems. All rights reserved.

Stylus Studio includes software developed by Saxonica Limited and is distributed on an "AS IS" basis WITHOUT WARRANTY OF ANY KIND, either express or implied. Copyright © 2005 Saxonica Limited. All rights reserved.

Stylus Studio includes software developed by the The Anti-Grain Geometry Project. Copyright © 2002-2005 Maxim Shemanarev (McSeem). This software is provided "as is" without express or implied warranty, and with no claim as to its suitability for any purpose.

September 2006

---

# Contents

<b>Preface</b>	v
About This Manual	v
Conventions in This Manual	v
Typographical Conventions	vi
Syntax Notation	vii
Information Alerts	vii
Video Alerts	vii
Video Descriptions	viii
Available Documentation	viii
 <b>Chapter 1: The adapter: URL Scheme</b>	 1
The adapter: URL Scheme	1
Invoking an Adapter URL in DataDirect XQuery	3
Stylus Studio Built-In Adapters	3
Adapter URL Syntax	4
Using Stylus Studio to Build an Adapter URL	4
Where Adapter URLs are Displayed in Stylus Studio	5
Using the URL in the Select XML Converter Dialog Box	6
Using the URL in the Properties Window	6
How to Open a File Using an Adapter in Stylus Studio	7
 <b>Chapter 2: Adapter Properties</b>	 11
Line Separator Values	12
Base-64 Adapter Properties	13
Adapter Name in URL	13

## Contents

---

Binary Adapter Properties . . . . .	14
Adapter Name in URL . . . . .	14
Comma-Separated and Tab-Separated Values Adapter Properties . . . . .	15
Adapter Names in URL . . . . .	15
dBase Adapter Properties . . . . .	17
Adapter Names in URL . . . . .	17
DIF, SDI, SLK Adapter Properties . . . . .	18
Adapter Names in URL . . . . .	18
EDI Adapter Properties . . . . .	19
Adapter Name in URL . . . . .	19
HtmlTidy Adapter Properties . . . . .	23
Adapter Name in URL . . . . .	23
Whole-line Text Adapter Properties . . . . .	24
Adapter Name in URL . . . . .	24
Properties for Other Adapters . . . . .	25
Notes on the MBox Adapter . . . . .	25
<b>Index . . . . .</b>	<b>27</b>

---

## Preface

This Preface contains the following sections:

- [About This Manual](#) describes this manual and its intended audience.
- [Conventions in This Manual](#) describes the text formatting, syntax notation, and flags used in this manual.
- [Available Documentation](#) describes the documentation that accompanies the Stylus Studio XML Deployment Components.

## About This Manual

This manual describes Stylus Studio XML Adapters and how to use them to access data in flat-file formats like CSV, EDI, dBase, and others.

This manual has the following chapters:

- [Chapter 1, “The adapter: URL Scheme,”](#) describes the adapter: URL scheme, Stylus Studio’s built-in adapters for CSV, EDI, and other file formats, and how to use Stylus Studio 2007 XML Enterprise Suite to build adapter: URLs.
- [Chapter 2, “Adapter Properties,”](#) describes values for the line separator property, which is common to most adapters, and it describes the properties for individual adapters.

## Conventions in This Manual

This section describes the typographical and formatting conventions used in this manual for text, notes, warnings, and important messages.

### Typographical Conventions

This manual uses the following typographical conventions:

- **Bold typeface in this font** indicates keyboard key names (such as **Tab** or **Enter**) and the names of windows, menu commands, buttons, and other user-interface elements. For example, “From the **File** menu, select **Open**.”
- *Italic text* emphasizes new terms when they are introduced.
- Code samples appear in text like this:

```
doc("adapter://EDI?ticker-request.edi")
```

- Monospace typeface indicates text that might appear on a computer screen such as
  - Code that the user must enter
  - System output (such as responses, error messages, and so on)
  - Filenames and pathnames
  - Software component names, such as class and method names

Essentially, monospace typeface indicates anything that the computer is “saying,” or that must be entered into the computer in a language that the computer “understands.”

**Bold monospace typeface** emphasizes text that would otherwise appear in monospace typeface.

*Monospace typeface in italics* or ***Bold monospace typeface in italics*** (depending on context) indicates variables or placeholders for values you supply or that might vary from one case to another.

◆ **Procedures are introduced this way:**



## Syntax Notation

This manual uses the following syntax notation conventions:

- Brackets ([ ]) in syntax statements indicate parameters that are optional.
- Braces ({} ) indicate that one (and only one) of the enclosed items is required. A vertical bar (|) separates the alternative selections.
- Ellipses (...) indicate that you can choose one or more of the preceding items.

## Information Alerts

This manual highlights special kinds of information by shading the information area, and indicating the type of alert in the left margin.

**Tip** A **Tip** flag identifies information that can help you use the software more effectively – short-cuts, alternatives, and information about system behavior are all examples of tips.

**Note** A **Note** flag indicates information that complements the main text flow. Such information is especially needed to understand the concept or procedure being discussed.

**Important** An **Important** flag indicates information that must be acted upon within the given context in order for the procedure or task (or other) to be successfully completed.

**Warning** A **Warning** flag indicates information that can cause loss of data or other damage if ignored.

## Video Alerts

Stylus Studio provides dozens of video demonstrations of editing tools and features for XSLT, XQuery, XML Schema, relational-to XML conversion, and others. Sections in the documentation that describe a Stylus Studio feature for which a video demonstration exists include an alert like the following:



Watch it! You can view a video demonstration of this feature by clicking the television icon or by clicking this link: [watch a video on using Stylus Studio with DataDirect XQuery](#).

Clicking either the television icon or the hyperlink launches the video.

### Video Descriptions

You can find descriptions of all Stylus Studio video demonstrations here:

[http://www.StylusStudio.com/xml\\_videos.html](http://www.StylusStudio.com/xml_videos.html)

## Available Documentation

Table 1 lists the documentation supplied with the Stylus Studio XML Deployment Components. In addition to the documentation listed in this table, the Stylus Studio XML Deployment Components come with sample files. All documentation is included with the Stylus Studio media and downloads.

**Table 1. Stylus Studio XML Deployment Components Documentation**

<i><b>Document</b></i>	<i><b>Description</b></i>
<i>Stylus Studio® 2007 XML Deployment Components Adapter URL Reference</i>	Provides reference information on the built-in adapter URL schemes supported by Stylus Studio.
<i>ReadMe</i>	Describes late-breaking information and known issues. The readme is located in the \doc directory where you installed Stylus Studio.
Javadoc	Javadoc for the Stylus Studio Java API is installed in the /doc/Javadoc folder where you installed the Stylus Studio XML Deployment Components. Open <code>index.html</code> to get started. Javadoc is also available on the Stylus Studio Web site, <a href="http://www.StylusStudio.com">http://www.StylusStudio.com</a> .

## Chapter 1    **The adapter: URL Scheme**

Stylus Studio uses the adapter: URL scheme extensively to reach a variety of data sources using both built-in adapters or, if you are also using Stylus Studio 2007 XML Enterprise Suite, user-defined converters.

This chapter describes the adapter: URL scheme and covers the following topics:

- “The adapter: URL Scheme”
- “Invoking an Adapter URL in DataDirect XQuery”
- “Stylus Studio Built-In Adapters”
- “Adapter URL Syntax”
- “Using Stylus Studio to Build an Adapter URL”
- “How to Open a File Using an Adapter in Stylus Studio”

### **The adapter: URL Scheme**

An adapter: URL that invokes the Stylus Studio built-in comma-separated values adapter to convert the three.txt file in the \examples\Adapters directory to XML document might look like this:

```
adapter:CSV:newline=crlf:sep=,:first=yes:escape=\"'?file:///c:/XMLDeploymentComponents/examples/Adapters/three.txt
```

The instructions to the adapter engine from this instance of the adapter URL are described in [Table 2](#).

**Table 2. Parts of an Adapter URL**

<i>Instruction</i>	<i>Adapter URL String</i>
Use the Comma-Separated Values adapter	adapter:csv
The line separator in the source file is a carriage return/line feed	newline=crlf
The column separator in the source file is a comma	sep=,
The values in the first row of the source file should be used to supply field names	first=yes
The escape character in the source file is a slash \	escape=\
The quote characters in the source file are " and '	quotes=""
The source file is three.txt	file:///c:/XMLDeploymentComponents/ examples/Adapters/three.txt

While the basic format of the adapter URL is the same from one adapter to another, there are differences. For example:

- Built-in adapters have different properties. For example, the adapter for dBase files has settings that the adapter for binary files does not. If you have Stylus Studio 2007 XML Enterprise Suite, you can use Stylus Studio to build your adapter URLs. See [“Using Stylus Studio to Build an Adapter URL”](#) for more information. Otherwise, you must construct them manually, taking care to specify both setting names and their values correctly. For a complete description of properties for all adapters, see [Chapter 2, “Adapter Properties”](#).
- The adapter: URL scheme can also be used to reference a user-defined converter (a .conv file). In this case, the adapter URL specifies only the location of the .conv file; the converter itself contains information about its property settings. An adapter URL that references a user-defined converter might look like this:

```
adapter:///myConverter.conv?file:inventory.txt
```

This adapter uses `myConverter.conv` to convert the file `inventory.txt` to some format (specified in the converter when you built it in Stylus Studio 2007 XML Enterprise Suite).

**Note**

User-defined converters can be defined using Stylus Studio 2007 XML Enterprise Suite only. They cannot be built programmatically, though they can be invoked programmatically.

## Invoking an Adapter URL in DataDirect XQuery

DataDirect XQuery uses the Stylus Studio URI resolver to interpret the `doc()` function. This enables the `doc()` function to take an `adapter: URL` as its argument. Consider the following example, which uses the `doc()` function to invoke the Stylus Studio CSV adapter to convert the file `request.csv` to XML:

```
doc("adapter://CSV:first=yes?request.csv")
```

In this example, only one of the CSV adapter properties is set (`first=yes`); for all other properties, default settings are used.

## Stylus Studio Built-In Adapters

Stylus Studio includes built-in adapters for the following file formats:

- Base-64
- Binary
- Comma-separated values (CSV)
- dBase
- Directory Interchange Format (DIF)
- Electronic Data Interchange (EDI)
- E-mail Mbox
- HtmlTidy
- Java .properties
- Rich Text Format (RTF)
- Single Document Interface (SDI)
- Symbolic Link (SLK)

- Tab-separated values (TSV)
- Whole-line text
- Windows .ini
- Windows Write

For a list of the properties associated with the Stylus Studio built-in adapters, see [Chapter 2, “Adapter Properties”](#).

## Adapter URL Syntax

While properties differ from one adapter to the next, the syntax used to invoke them is the same:

`adapter:name:[property_name=value: | property_name=value: | ...]?file:file URL`

Example:

`adapter:Base64:newline=crLf:encoding=utf-8?file//w:\myfiles\base_to_xml.bin`

In this example:

- The name of the adapter is Base64
- Properties specified are `newline` and `encoding`; default values are used for all other adapter properties
- The file being converted is `base_to_xml.bin` on `w:\myfiles`

## Using Stylus Studio to Build an Adapter URL

If you have Stylus Studio 2007 XML Enterprise Suite, you can use Stylus Studio to construct the adapter URLs you use in Java application. Adapter URLs can be complex – properties and their values vary from one adapter to another, for example – and long, so using Stylus Studio to construct them can reduce errors in your applications.

This section covers the following topics:

- [“Where Adapter URLs are Displayed in Stylus Studio”](#)
- [“Using the URL in the Select XML Converter Dialog Box”](#)
- [“Using the URL in the Properties Window”](#)

## Where Adapter URLs are Displayed in Stylus Studio

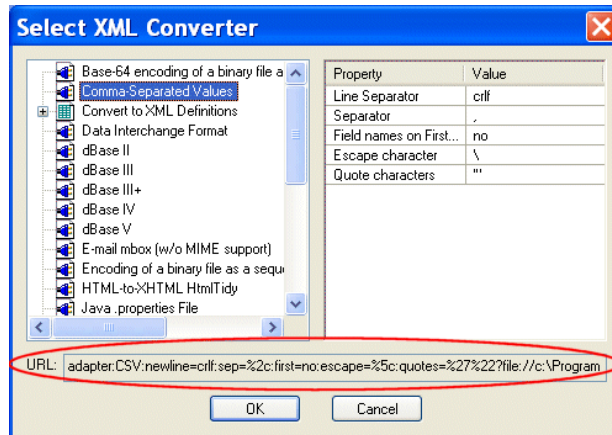
If you are using Stylus Studio 2007 XML Enterprise Suite, you can view adapter URLs

- In the **Project** window (select **Show Full URL** from the **Project** window shortcut menu)



**Figure 1. Adapter URL Displayed in Project Window**

- In the **URL** field of the **Select XML Converter** dialog box, as shown in [Figure 2](#).



**Figure 2. Adapter URL Displayed in the Select XML Converter Dialog Box**

You can use either source for the adapter URL strings in your Java applications.

## Using the URL in the Select XML Converter Dialog Box

**To construct an adapter URL using the URL in the Select XML Converter dialog box:**

1. Use the adapter to open a file as an XML document in Stylus Studio. This can be a user-defined converter or a built-in adapter, as appropriate. See [“How to Open a File Using an Adapter in Stylus Studio”](#) if you need help with this step.
2. Before clicking **OK** to complete the conversion, copy the adapter URL in the **URL** field of the **Select XML Converter** dialog box (see [Figure 2](#)).

**Tip**

Adapter URLs in the **Select XML Converter** dialog box are already escaped and can be used as-is. Adapter URLs taken from the **Project** window must be escaped manually when you paste them into your application.

3. Click **OK** to complete the conversion. (You can click **Cancel** if you are performing this procedure just to obtain the adapter URL.)
4. Paste the adapter URL into your Java program.

## Using the URL in the Properties Window

**To construct an adapter URL using the URL in the Properties window:**

1. Use the adapter to open a file as an XML document in Stylus Studio. This can be a user-defined converter or a built-in adapter, as appropriate. See [“How to Open a File Using an Adapter in Stylus Studio”](#) if you need help with this step.

**Tip**

New documents are placed in the **Other Documents** folder in the **Project** window by default.

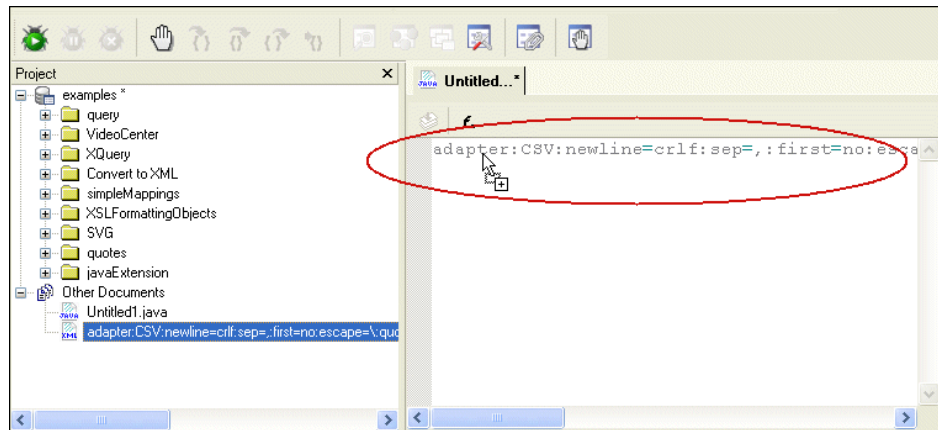
2. Open a new document in any Stylus Studio text editor (for example, **File > New > Java: Text Editor**).

The purpose of this step is to provide an editor into which you can drag-and-drop the document you created in [Step 1](#) in order to display the associated adapter URL.

3. Drag the document you created in [Step 1](#) from the **Project** window and drop it into the text editor you opened in [Step 2](#).



The complete URL appears in the text editor.



**Figure 3. Copying a URL to a Text Editor**

4. Copy the complete adapter URL.
5. Paste the adapter URL into your Java program.

**Note**

Escape characters as required for strings in Java programs. For example, `escape=\":quotes='\"` becomes `escape=\\\":quotes='\"` (the single quote does not need to be escaped).

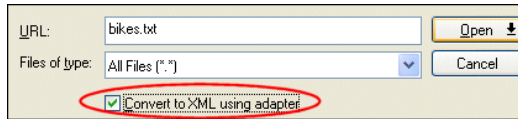
## How to Open a File Using an Adapter in Stylus Studio

You can use the **Open** dialog box to open a file using a user-defined converter or a built-in adapter in Stylus Studio. In the context of writing Java applications to invoke an adapter, you use this procedure to display the adapter: URL scheme, as described in [“Using Stylus Studio to Build an Adapter URL”](#).

**To open a file using an adapter from the Open dialog box:**

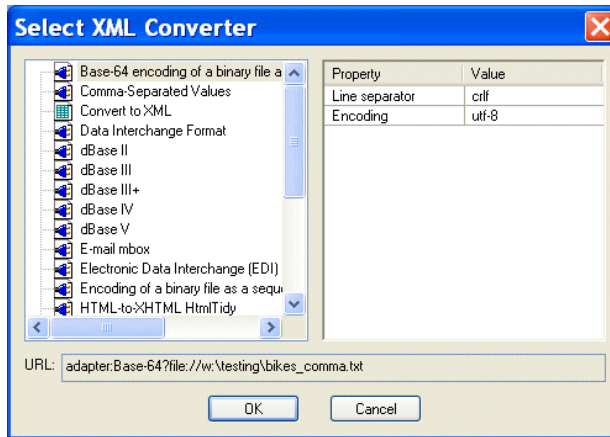
1. Display the **Open** dialog box (select **File > Open** from the Stylus Studio menu, for example).
2. Navigate to the directory that contains the file you want to open using the adapter and select the file.

3. Check the **Convert to XML using adapter** check box and click the **Open** button.



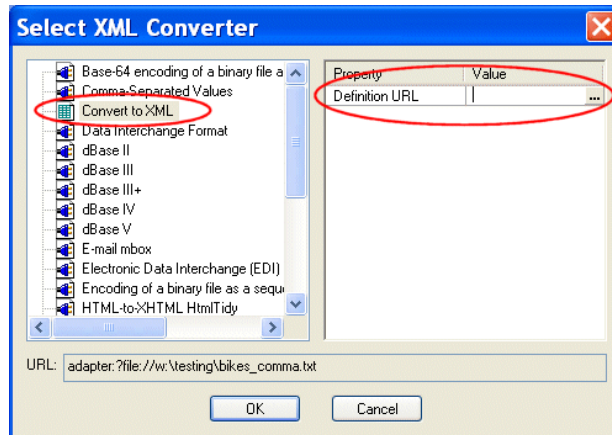
**Figure 4. Check Box to Open Files Using Adapter**

Stylus Studio displays the **Select XML Converter** dialog box.



**Figure 5. Select XML Converter Dialog Box**

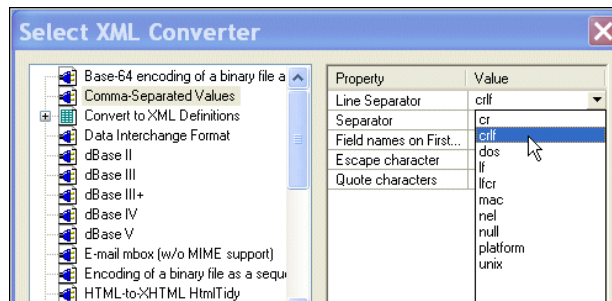
4. If you are using a user-defined converter, expand the **Convert to XML Definitions** tree and select the converter you want to use to convert the file you selected in [Step 2](#) to XML and go to [Step 6](#).



**Figure 6. Selecting a User-Defined Converter**

If you are using a built-in adapter, select the adapter you want to use and go to [Step 5](#).

5. Specify the values you want to use for translation purposes. Many fields have a drop-down list that displays valid values, like the **Line Separator** property for the Comma-Separated Values converter shown here:



**Figure 7. Built-In Converters Display Valid Properties**

See [Chapter 2, “Adapter Properties”](#) for information on properties for specific adapters.

6. Click **OK**.

## The adapter: URL Scheme

---

The file is converted to XML and appears in the editor from which you displayed the **Open** dialog box in [Step 1](#).

**Tip** You can use the same basic procedure, from the **Save As** dialog box, to save a file using an adapter.

## **Chapter 2**    **Adapter Properties**

Stylus Studio built-in adapters share some properties in common (the line separator property, for example), and they all have properties that are unique to them – the CSV adapter allows you to specify an escape character, but the binary adapter does not, for example.

This chapter describes values for the line separator property, which is common to most adapters, and it describes the properties for individual adapters.

This chapter covers the following topics:

- “Line Separator Values”
- “Base-64 Adapter Properties”
- “Binary Adapter Properties”
- “Comma-Separated and Tab-Separated Values Adapter Properties”
- “dBase Adapter Properties”
- “DIF, SDI, SLK Adapter Properties”
- “EDI Adapter Properties”
- “HtmlTidy Adapter Properties”
- “Whole-line Text Adapter Properties”
- “Properties for Other Adapters”

# Line Separator Values

Most adapters allow you to specify some type of line separator (referred to in the adapter URL as *newline*). The following table summarizes commonly occurring values. All values are case-insensitive.

**Table 3. Common Values for the Line Separator (newline) Property**

<i><b>Value</b></i>	<i><b>Description</b></i>
cr or mac	The Macintosh standard.
lf or unix	The Unix standard.
crlf or dos	The DOS standard.
lfcr	The Windows standard.
ne1	0x85 (commonly found in mainframes).
null	A null byte.
platform	If another value has not been specified, the line separator uses the platform value as returned by the <code>System.getProperty("line.separator")</code> method.

## Base-64 Adapter Properties

The following table shows adapter properties for Base-64 encoded binary files as documented in RFC 1341.

### Adapter Name in URL

Base-64

**Table 4. Properties for Base-64 Adapters**

<i>Property Name</i>	<i>Name in URL</i>	<i>Description</i>
Line separator	newline	Used only when converting a Base-64 binary file to XML, and not vice versa. The default is crlf. See <a href="#">“Line Separator Values”</a> for a list of values.
Encoding	encoding	The encoding for the input file when it is not XML; or the encoding for the output file when it is not XML. The default is utf-8.

# Binary Adapter Properties

You can convert binary files that have been encoded as a sequence of digits in a base from 2 to 36, and vice versa. Use the Base-64 adapter for base-64 encoded binary files. See [“Base-64 Adapter Properties”](#) for more information.

## Adapter Name in URL

Binary

**Table 5. Properties for Binary Base-2 to Base-36 Adapters**

<i><b>Property Name</b></i>	<i><b>Name in URL</b></i>	<i><b>Description</b></i>
Line separator	newline	Used when converting a binary encoded file to XML, and vice versa. The default is crlf. See <a href="#">“Line Separator Values”</a> for a list of values.
Encoding	encoding	The encoding for the input file when it is not XML; or the encoding for the output file when it is not XML. The default is utf-8.
Base	base	The numeric base of the encoded file. The default is 16 (hexadecimal). Base-2 is binary; base-8 is octal; and base-10 is decimal.
Wrap lines	wrap	Whether you want to wrap lines (wrap=yes) or output all values on a single line (wrap=no).
Byte separator	space	Whether or not byte values should be contiguous (no value) or separated with the value specified for this property. For example, if you set space=, the value 000FFF would be output as 00,0F,FF.



## Comma-Separated and Tab-Separated Values Adapter Properties

With a few exceptions, noted in the following table, adapter properties are the same for both the comma-separated values and tab-separated values adapters.

### Adapter Names in URL

CSV (comma-separated values)

TAB (tab-separated values)

**Table 6. Properties for CSV and TSV Adapters**

<b><i>Property Name</i></b>	<b><i>Name in URL</i></b>	<b><i>Description</i></b>
Encoding	encoding	The encoding for the input file when it is not XML; or the encoding for the output file when it is not XML.
Line separator	newline	See <a href="#">“Line Separator Values”</a> for a list of values.
Separator	sep	The separator value between each value. This can be 'TAB', any single character (a comma ( , ) is the default), or the %XX-escaped value (%2c, for example).
First row contains field names	first	Generated field names depend on the values in the first and number fields.  If first=yes and number=no, field names are read from the first row. Any field names after that are named column.xxx, where xxx is the column number, starting from one and including explicitly named columns in the count. If number=yes, extra columns (those after the first) are named just column.
Escape character	escape	This character escapes quotes and separators so that they can be embedded in values. The backslash (\) is the default.

**Table 6. Properties for CSV and TSV Adapters**

<b><i>Property Name</i></b>	<b><i>Name in URL</i></b>	<b><i>Description</i></b>
Quote character	quote	A list of characters the adapter should interpret as quotation characters. Double and single quote marks (" ") are the default values.
Number rows and columns	number	<p>If number=yes (no is the default), each row will also have an attribute, named row, which will contain the row number from the source document, starting from one. Also, each column, even those explicitly named, will have a column attribute numbering the column from one.</p> <p>Any empty columns are omitted from the output, but the numbering of subsequent columns will reflect that a column(s) was skipped.</p>

## dBase Adapter Properties

Adapter properties are the same for all dBase adapters – dBase II, dBase III, dBase III+, dBase IV, and dBase V.

### Adapter Names in URL

dBase\_II

dBase\_III

dBase\_III\_plus

dBase\_IV

dBase\_V

**Table 7. Properties for dBase Adapters**

<i>Property Name</i>	<i>Name in URL</i>	<i>Description</i>
Line separator	newline	Used only to convert a dBase file to XML, not vice versa. See <a href="#">“Line Separator Values”</a> for a list of values.
Encoding	encoding	The encoding for the input file when it is not XML; or the encoding for the output file when it is not XML. The default is utf-8.
Include deleted records	deleted	Whether or not records marked with a "deleted" attribute are included in the output to XML and preserved in the conversion from XML. Stylus Studio generates the "deleted" attribute on output, and looks for it on input when this property is set to Yes.

# DIF, SDI, SLK Adapter Properties

Adapter properties are the same for both Data Interchange Format (DIF), Super Data Interchange Format (SDI), and Symbolic Link Format (SYLK) adapters.

## Adapter Names in URL

DIF (Data Interchange Format)

SDI (Super Data Interchange Format)

SYLK (Symbolic Link Format)

**Table 8. Properties for DIF, SDI, and SYLK Adapters**

<i><b>Property Name</b></i>	<i><b>Name in URL</b></i>	<i><b>Description</b></i>
Line separator	newline	Used when converting DIF, SDI, or SYLK file to XML, and vice versa. The default is crlf. See <a href="#">“Line Separator Values”</a> for a list of values.
Encoding	encoding	The encoding for the input file when it is not XML; or the encoding for the output file when it is not XML. The default is utf-8

## EDI Adapter Properties

Adapter properties are the same for both Data Interchange Format (DIF), Super Data Interchange Format (SDI), and Symbolic Link Format (SLK) adapters.

### Adapter Name in URL

EDI

**Table 9. Properties for EDI Adapters**

<b>Property Name</b>	<b>Name in URL</b>	<b>Description</b>
Line separator	newline	Used when converting EDI to XML, and XML to EDI when the <i>Add linefeeds between segments on write</i> property (eo1) is set to yes. The default is crlf. See <a href="#">“Line Separator Values”</a> for a list of values.
Encoding	encoding	The encoding for the input file when it is not XML; or the encoding for the output file when it is not XML. The default is utf-8.
Enable validation	val	Validates the XML against the structure in the relevant EDI dictionary. An error is generated if the EDI dialect (EDIFACT, X12, for example) isn’t recognized; an error is also generated if the dialect is recognized, but the message type isn’t. Missing mandatory segments, or segments not specified for a particular group will also generate errors. Consider leaving this option on (Yes, the default). If this option is off (No), the adapter is unable to synchronize its position within the EDI dictionary, preventing nested loops from being generated with the output. When possible, leave this property set to Yes and ensure that the EDI input conforms to the specification.

**Table 9. Properties for EDI Adapters**

<b>Property Name</b>	<b>Name in URL</b>	<b>Description</b>
Comment code list data	decode	<p>Adds to each code that is looked up in a table a comment that explains the code's value. For example, <code>&lt;!--Production Data--&gt;</code> in the following code:</p> <pre>&lt;ISA15&gt;&lt;!--I14: Interchange Usage Indicator--&gt;P&lt;!--Production Data--&gt;&lt;/ISA15&gt;</pre> <p>Turn off this and <i>Comment element types</i> (field) to disable all comment generation.</p>
Comment element types	field	<p>Creates a comment at the start of each element that includes the element's name and number. For example, <code>&lt;!--I14: Interchange Usage Indicator--&gt;</code> in the following code:</p> <pre>&lt;ISA15&gt;&lt;!--I14: Interchange Usage Indicator--&gt;P&lt;!--Production Data--&gt;&lt;/ISA15&gt;</pre> <p>Turn off this and <i>Comment code list</i> (decode) to disable all comment generation.</p>
Strict validation on value lengths	len	Checks each value against the upper and lower length limits defined in the EDI specification.
Strict segment-ordering checking	seg	Relaxes the rules that require that segments come in the specified order. However, if this property is off (No), some looping constructs might break, resulting in data being grouped in correctly.
Force error if value not in code list	tbl	Generates an error if the value for an element is not in the codelist associated with that element. If this property is off (No), values are not checked for the presence of a codelist.

**Table 9. Properties for EDI Adapters**

<b>Property Name</b>	<b>Name in URL</b>	<b>Description</b>
Strict datatype content checking	typ	Ensures that only characters that are appropriate for a given field are included in the value for that field. For example, this property ensures that dates are valid and numbers are well-formed.
Treat all segments as optional	opt	<p>If set to Yes (No is the default), Stylus Studio assumes that all segments are optional. This property can be useful if your provider declines to provide segments that are considered mandatory according to the EDI specification, but you are aware of what the missing values are.</p> <p>This property is not used if <i>Enable validation</i> (val) is set to Yes.</p>
Add linefeeds between segments on write	eol	Allows you to put each segment on its own line when converting XML to EDI. (Extra linefeeds are ignored when converting EDI to XML.) If this property is set to Yes (the default), the value specified in the Line separator (newline) property is used to separate each segment. The normal segment output character is also generated.
Namespace prefix	prefix	Namespace prefix to be added, with the <i>Namespace URI</i> , to the root element. The prefix alone is added to all elements.

**Table 9. Properties for EDI Adapters**

<b>Property Name</b>	<b>Name in URL</b>	<b>Description</b>
Namespace URI	uri	Namespace URI to be added, with the <i>Namespace prefix</i> , to the root element. If the prefix is set, but the URI is not, the prefix is ignored.
Loop element name	loop	<p>Used to change the group element name. By default, segment loops are wrapped in &lt;GROUP_#&gt; elements, where # denotes the segment group number. The loop element name can be any valid element name, with the following exceptions for these characters:</p> <p># – the segment group number is used instead.</p> <p>\$ – the name of the first segment of the group is inserted into the element name instead.</p> <p>@ – the portion of the loop element name before the @ symbol is used as the element name. If nothing occurs before the @ symbol, GROUP is used, and the portion after, if any, is used as the attribute name. The attribute value is the group number or the first segment name within the group, depending on whether a # or \$ is within the attribute name.</p>



# HtmlTidy Adapter Properties

You can use the HtmlTidy adapter to convert HTML to XHTML.

## Adapter Name in URL

HTMLTidy

**Table 10. Properties for the HtmlTidy Adapter**

<i><b>Property Name</b></i>	<i><b>Name in URL</b></i>	<i><b>Description</b></i>
Line separator	newline	Used when converting HTML to XHTML. The default is crlf. See <a href="#">“Line Separator Values”</a> for a list of values.
Encoding	encoding	The encoding for the input file when it is not XML; or the encoding for the output file when it is not XML. The default is utf-8.
Abort on warnings found	warnings	Whether or not you want the adapter to fail when it encounters potential problems with the HTML-to-XHTML mapping. warnings=no is the default.
Abort on errors found	errors	Whether or not you want the adapter to fail when it encounters problems with the HTML-to-XHTML mapping. If errors=no, the adapter continues with the conversion, making a best guess. errors=yes is the default.

# Whole-line Text Adapter Properties

The following table describes the properties for the Whole-line Text (Line) adapter.

## Adapter Name in URL

Line

**Table 11. Properties for the Whole-line Text Adapter**

<i><b>Property Name</b></i>	<i><b>Name in URL</b></i>	<i><b>Description</b></i>
Line separator	newline	Used when converting a whole-line text file to XML, and vice versa. The default is crlf. See <a href="#">“Line Separator Values”</a> for a list of values.
Encoding	encoding	The encoding for the input file when it is not XML; or the encoding for the output file when it is not XML. The default is utf-8.
Root element name	root	Value used for the root element name. Default is root.
Line element name	line	Value used for the line element name. Default is line.

## Properties for Other Adapters

Each of the following adapters has the same properties. Names as they appear in adapter URLs are shown following the adapter's proper name.

- E-mail Mbox (MBox)
- Java .properties File (JavaProps)
- Progress .d Data Dump (DotD)
- Rich Text Format (RTF)
- Windows .ini File (WinIni)
- Windows Write (WinWrite)

**Table 12. Properties for Other Adapters**

<i><b>Property Name</b></i>	<i><b>Name in URL</b></i>	<i><b>Description</b></i>
Line separator	newline	Used when converting a file to XML, and vice versa. The default is crlf. See <a href="#">“Line Separator Values”</a> for a list of values.
Encoding	encoding	The encoding for the input file when it is not XML; or the encoding for the output file when it is not XML. The default is utf-8.

### Notes on the MBox Adapter

The E-mail MBox (MBox) adapter can be used for MBox-to-XML conversion only. This adapter supports MIME attachments – any message attachments are properly decoded and rendered as XML. Specifically:

- Text and plain attachments are embedded as text
- Binary attachments are embedded as hex-encoded data
- XML-encoded attachments are emitted as escaped text



---

# Index

## A

- adapter URL scheme
  - building your own adapter URL 4
  - displayed in Stylus Studio 5
  - parts of 1
  - syntax of 4
  - using with user-defined .conv files 2
- adapters
  - Base-64 adapter properties 13
  - binary adapter properties 14
  - building an adapter URL 4
  - built-in adapter types 3
  - comma-separated value adapter properties 15
  - CSV adapter properties 15
  - dBase adapter properties 17
  - DIF adapter properties 18
  - DotD adapter properties 25
  - EDI adapter properties 19
  - file formats supported by built-in adapters 3
  - HtmlTidy adapter properties 23
  - JavaProps adapter properties 25
  - line separator values used in 12
  - MBox adapter properties 25
  - opening files with adapters in Stylus Studio 7
  - properties for built-in adapters 11
  - RTF adapter properties 25
  - SDI adapter properties 18
  - SLK adapter properties 18
  - tab-separated value adapter properties 15
  - Whole-line Text adapter properties 24

- WinIni adapter properties 25
- WinWrite adapter properties 25

## B

- Base-64 adapter properties 13
- binary adapter properties 14

## C

- comma-separated value adapter properties 15
- CSV adapter properties 15

## D

- dBase adapter properties 17
- DIF adapter properties 18
- documentation
  - Stylus Studio documentation viii
- DotD adapter properties 25

## E

- EDI adapter properties 19

## F

- file formats

built-in adapter support for 3

## H

HtmlTidy adapter properties 23

## J

JavaProps adapter properties 25

## M

Mbox adapter properties 25

## R

RTF adapter properties 25

## S

SDI adapter properties 18

SLK adapter properties 18

Stylus Studio

- building an adapter URL using 4

- opening files with an adapter 7

syntax, notations used in this manual vii

## T

tab-separated value adapter properties 15

typographical conventions vii

## U

user-defined converters

- using with adapter URLs 2

## V

video demonstrations

- alerts for vii

documentation alerts for vii

## W

Whole-line Text adapter properties 24

WinIni adapter properties 25

WinWrite adapter properties 25