

XEENA

XML Editing Environment, Naturally in Java!

(C) 1999-2000 IBM Corporation. All rights reserved.

XEENA is brought to you by the IBM Haifa Research Lab, Israel.
Please send comments to helpmap@il.ibm.com
and check our web site on alphaWorks
<http://www.alphaWorks.ibm.com/tech/xeena>

1. [What is Xeena?](#)
 2. [Why should I use Xeena?](#)
 3. [Does Xeena support the full XML specification?](#)
 4. [To what extent does Xeena validate the files it creates/edits?](#)
 5. [How do I:](#)
 1. [Configure Xeena for my grammar ?](#)
 2. [Configure Xeena to enable importing documents from different formats?](#)
 3. [Configure Xeena to enable exporting documents into different formats?](#)
 4. [Configure Xeena to enable invoking a tool on a document?](#)
 5. [Configure Xeena to use a different UI Look&Feel?](#)
 6. [Configure Xeena to use an HTTP proxy?](#)
-

1. What is Xeena?

Xeena - XML Editing Environment, Naturally in Java.

Editing Environment - as it can take a DTD/Schema, and lets you create and edit XML files that conform to it.
Naturally in Java - so you can run it on any platform.

Xeena features:

- Intuitive viewing and editing of XML documents in a tree control view.
- Editing of multiple XML documents.
- Includes XML source viewer.
- Restricts adding and editing of features according to the DTD/Schema, and checks validity of produced documents during editing.
- Invalid elements in the document are underlined in red.
- Easy customization of display.
- Hooks for importing/exporting documents, and invoking tools on documents.

2. Why should I use Xeena?

If you're looking for a way to create and edit XML files without having to learn the intricacies of XML - Xeena is for you.

3. Does Xeena support the full XML specification?

Xeena supports the XML specification specified in <http://www.w3.org/TR/1998/REC-xml-19980210>, except for the following:

- DTD inside XML files.
- Entity references (Note that character references are allowed in attribute values and generated automatically by the editor).

4. To what extent does Xeena validate the files it creates/edits?

Xeena ensures validity of the structure of each node in the tree (e.g. it will not let the user insert an element in a place which is not allowed by the DTD/Schema). Xeena also checks validity of the data associated with each node (e.g. attributes values for elements, target value for Processing Instructions).

Invalid nodes are underlined in red immediately after opening a file and are checked constantly during editing.

Selecting the invalid node and activating the "Validate" toolbar/menu action prints the error messages associated with that node in the message area.

5. How do I:

1. Configure Xeena for my grammar (DTD/Schema)?

See the [configuration instructions](#) in the help file.

2. Configure Xeena to enable importing documents from different formats?

Xeena can be configured to import non-XML files into XML documents, by invoking Java code you supply to implement the transformation. To do this, you need to:

- Write a Java class that implements the [XMLImporter](#) interface. This consists of 3 main steps: implement the [getIcon](#), [getName](#) and [getToolTip](#) methods to return descriptions of your importer, to be used by the Xeena environment; implement the [init](#) methods to initialize your importer (using information from the [XMLEditorContext](#) and/or the data string you may have specified in the configuration file; implement the [importDocument](#) method to do the actual work of letting the user select an import source, then importing the data into the document. For a detailed sample, please examine the [BMLImporter.java](#) sample supplied in the src directory.
- Hook the class you wrote into the Xeena profile for your DTD/Schema: when configuring Xeena for your DTD/Schema, insert a new *importer* element (as a child of the existing *importers* element), specify its name, and specify the full name of the class you wrote as the value of the *class* attribute.

3. Configure Xeena to enable exporting documents into different formats?

Xeena can be configured to export XML documents into non-XML files, by invoking Java code you supply to implement the transformation. To do this, you need to:

- Write a Java class that implements the [XMLExporter](#) interface. This consists of 3 main steps: implement the [getIcon](#), [getName](#) and [getToolTip](#) methods to return descriptions of your exporter, to be used by the Xeena environment; implement the [init](#) methods to initialize your exporter (using information from the [XMLEditorContext](#) and/or the data string you may have specified in the configuration file; implement the [exportDocument](#) method to do the actual work of letting the user select an export target, then exporting the data from the document. For a detailed sample, please examine the [BMLExporter.java](#) sample supplied in the src directory.
- Hook the class you wrote into the Xeena profile for your DTD/Schema: when configuring Xeena for your DTD/Schema, insert a new *exporter* element (as a child of the existing *exporters* element), specify its name, and specify the full name of the class you wrote as the value of the *class* attribute.

4. Configure Xeena to enable invoking a tool on a document?

Xeena can be configured to apply a tool to XML documents, by invoking Java code you supply. To do this, you need to:

- Write a Java class that implements the [ToolHandler](#) interface. This consists of 4 main steps: implement the [getIcon](#), [getName](#) and [getToolTip](#) methods to return descriptions of your tool, to be used by the Xeena environment; implement the [init](#) methods to initialize your tool (using information from the [XMLEditorContext](#) and/or the data string you may have specified in the configuration file; implement the [canHandle](#) to give an indication whether the selected document can be handled by the tool; implement the [handle](#) method to do the actual work of invoking the tool on the selected document. For a detailed sample, please examine the [BMLPlayer.java](#) sample supplied in the src directory.
- Hook the class you wrote into the Xeena profile for your DTD/Schema: when configuring Xeena for your DTD/Schema, insert a new *tool* element (as a child of the existing *tools* element), specify its name, and specify the full name of the class you wrote as the value of the *class* attribute.

5. Configure Xeena to use a different UI Look&Feel?

Edit the xeena.bat (Windows) or xeena.sh (Unix) file in the root directory of your Xeena installation and change the name of the Look and Feel class (look for the string

-Dswing.defaultlaf=com.sun.java.swing.plaf....).

Alternatives you can use are:

- com.sun.java.swing.plaf.windows.WindowsLookAndFeel
- com.sun.java.swing.plaf.mac.MacLookAndFeel
- com.sun.java.swing.plaf.motif.MotifLookAndFeel

If you remove the option altogether, the default Look and Feel will be used.

6. Configure Xeena to use an HTTP proxy?

If you intend to use Xeena to access DTDs or XML files which reside on external (outside your firewall) servers, you may need to tell Xeena which proxy server to use. This can be done via the InstallShield wizard (Windows only), or thereafter by editing the `xeena.bat` (Windows) or `xeena.sh` (Unix) file, and setting the `PROXY_SETTINGS` environment variable (by default set to "`-DproxySet=false`") to:

```
"-DproxySet=true -DproxyHost=proxy.server.name -DproxyPort=port "
```

where `proxy.server.name` is the name of your proxy server, and `port` is the port number.

XEENA

XML Editing Environment, Naturally in Java!

(C) 1999-2000 IBM Corporation. All rights reserved.

XEENA is brought to you by the IBM Haifa Research Lab, Israel.
Please send comments to helpmap@il.ibm.com
and check our web site on alphaWorks
<http://www.alphaWorks.ibm.com/tech/xeena>

IMPORTANT - PLEASE READ THE INSTALLATION INSTRUCTIONS BEFORE RUNNING THE PROGRAM!

The XEENA Early Access release (Xeena-EA) is our way of delivering an xml editor with experimental schema support functionality.

XEENA-EA will comprise a series of ongoing updates -- Watch alphaWorks for new updates. We expect that functionality from Xeena-EA will migrate into a stable version of Xeena, but it's form in the stable version may (and in the cases of developing standards, definitely will) be different from it's form in Xeena-EA.

Please see the [Changes document](#) for a list of changes since the last version, and the [FAQ document](#) for Frequently Asked Questions.

1. [Included packages](#)
2. [Requirements](#)
3. [Installation](#)
 1. [Windows 95/98/NT](#)
 2. [Unix](#)
4. [How to use ?](#)
5. [How to test ?](#)
 1. [Windows 95/98/NT](#)
 2. [Unix](#)
6. [Known problems](#)
7. [Acknowledgments](#)

Included packages

Xeena includes part of the XML Parser Xerces (version 1.0.3) to parse xml/dtd/schema files. More information about Xerces can be obtained from [apache](#). Bug reports, requests for features, etc. pertaining to Xerces should be referred directly to the Xerces team. Xeena has not been tested with other versions of Xerces.

Xeena includes part of the XSL Processor Xalan (version 1.0.1) to implement the sample XSL tool. More information about Xalan can be obtained from [apache](#). Bug reports, requests for features, etc. pertaining to Xalan should be referred directly to the Xalan team. Xeena has not been tested with other versions of Xalan, and it is included solely for demonstration purposes.

To demonstrate the importing/exporting capabilities of Xeena, we have included hooks to the IBM Bean Markup Language (BML) toolkit (version 2.4). See [detailed instructions](#) in the Help file. More information about BML can be obtained from [alphaWorks](#). Bug reports, requests for features, etc. pertaining to BML should be referred directly to the BML team via the [BML discussion list](#) on alphaWorks. Xeena has not been tested with other

versions of BML, and it is referred to solely for demonstration purposes.

Requirements

Xeena is implemented in Pure Java, so it can run on any platform that has a Java virtual machine, with JDK version 1.1.6 and up and jdk 1.2.2. If you don't have any of these JDK versions on your computer, you can download it from [JavaSoft](#).

Installation

Windows95, WindowsNT (InstallShield)

- Install Xeena by running the Xeena-1.2EA.exe file

Unix

- Extract the distribution package Xeena-1.2EA.tar.zip:
unzip Xeena-1.2EA.tar.zip
tar -xvf Xeena-1.2EA.tar

How to use ?

Xeena can be used in two modes:

- Normal mode: as an XML editor which creates/edits files that conform to a specific dtd/schema. In order to start Xeena in this mode type:

```
xeena -dtd "dtd-url" [-root "r1,r2,..."] [-xml "xml-url"] [-profile cfg-url]
```

or

```
xeena -xsd "xml-schema-url" [-root "r1,r2,..."] [-xml "xml-url"] [-profile cfg-url]
```

Description

-dtd dtd-url

DTD file name.

-xsd xml-schema-url

DTD file name.

-root root1,root2,...

List of element names declared in the dtd/schema. This option specifies the types of documents Xeena supports. If not specified, any element can be a root.

-xml xml-file

If specified Xeena starts and immediately opens the xml-file.

-profile cfg-url

DTD/Schema configuration url name.

If specified it is used as the profile of the dtd/schema otherwise

Xeena looks for a dtd/schema-name.profile file in your USER_HOME

or in the same directory as the dtd/schema file.

- Customizing mode:
Xeena can be customized in two levels: **general**, which affects all invocations of Xeena, and **grammar-specific**, which is specific to a particular grammar. When customizing at the general level, you can change the icons used in the actions toolbar, as well as the color scheme Xeena will use. When customizing at the grammar-specific level, you can specify the icons used for the grammar elements, which tooltips appear for elements and attributes, how attributes are edited, if the

sidebar/toolbar palettes should be displayed and what elements the displayed palettes contain. The customization information should be contained in two files: *xeena.profile* (general mode) and *<dtd-name>.profile* (grammar-specific mode). You can use Xeena itself in order to create these files:

- To start Xeena in general mode type:

Xeena -cfg [-xml "xml-file"]

- To start Xeena in grammar specific mode type:

Xeena -cfg -dtd "dtd-url" [-root "r1,r2,..."] [-xml "xml-url"]

or

Xeena -cfg -xsd "xml-schema-url" [-root "r1,r2,..."] [-xml "xml-url"]

Description

-dtd dtd-url

DTD url.

-xsd xml-schema-url

XML Schema url.

-root r1,r2,...

List of element names declared in the dtd. This option specifies the types of documents Xeena supports. If not specified, any element can be a root.

-xml xml-url

If specified Xeena starts and immediately opens the xml-file.

You should put these files either in your USER_HOME or in the same directory as your grammar. Your USER_HOME is determined by Xeena according to your environment settings. In case you don't know what it is type:

Xeena -help

The following message is printed (describing in short the above). Look at the section which describes the -cfg flag and check what is your USER_HOME as found by Xeena.

usage:

xeena [-help] -dtd dtd-url [-root root1,root2,...] [-xml xml-url] [-profile profile-url]

or

xeena [-help] -xsd xml-schema-url [-root root1,root2,...] [-xml xml-url] [-profile profile-url]

or

xeena [-help] -cfg [-dtd|-xsd dtd-or-schema-url [-root root1,root2,...]] [-xml xml-url]

Description :

-help

Print out this message.

-dtd dtd-url

DTD url name.

-xsd schema-url

XML Schema url name.

-root root-name1,root-name2,...

List of element names declared in the dtd/schema. This option

specifies the types of documents Xeena supports.
If not specified, any element can be a root.

-xml xml-url

If specified Xeena starts and immediately opens the xml-url.

-cfg

Invoke Xeena for creating profile files (optional).

Use this option in order to customize Xeena either for a specific dtd/schema (if the option -dtd|-xsd grammar-file is specified as well) or for every dtd/schema.

The profile files should be named xeena.profile and dtd-name.profile and be located either in your USER_HOME directory (first place Xeena looks for the files) which was found to be

<USER_HOME>

or in the same directory as the dtd/schema file.

-profile profile-url

DTD/Schema profile url name.

If specified it is used as the profile of the dtd/schema otherwise

Xeena looks for a dtd/schema-name.profile file in your USER_HOME

(<USER_HOME>) or

in the same directory as the dtd/schema file.

When xeena is invoked in normal mode it looks for these files and if found Xeena customizes itself accordingly.

How to test ?

Windows95, WindowsNT

From the Start menu choose:

- **Xeena->Addressbook DTD example**
Xeena will start using the sample address book DTD.
- **Xeena->Addressbook Schema example**
Xeena will start using the sample address book XML Schema.
- **Xeena->Schema Editor**
Xeena will start using the W3C XML schema dtd allowing you to create/edit/view XML schema documents. Open addressbook.xsd to view the Addressbook sample schema. Use the menu option File->import->DTD in order to import any DTD document into an XML schema document. The XML Schema editor uses the 17 December 1999 Working Drafts for [Structures](#) and [Datatypes](#) (the version supported by the XML parser Xeena uses). The files structures.dtd datatypes.dtd and structures.profile reside in the samples directory.
- **Xeena->Config Addressbook DTD**
Xeena will start in customizing mode allowing you to create a profile file for the address book DTD (we already created such a file which is opened automatically).
- **Xeena->Config Xeena**
Xeena will start in customizing mode allowing you to create a profile file for Xeena itself. Xeena starts with an opened profile file (located in the same directory as the addressbook DTD). This was achieved by using the flag "-xml" (see [configxeena.bat](#)).

You can check the files [sample_dtd.bat](#), [sample_xsd.bat](#), [configdtd.bat](#) and [configxeena.bat](#), [editschema.bat](#) to see the command line in each mode (located under the directory where you installed xeena).

If you're running Windows 95, and encounter an "Out of environment space" error, then you'll need to increase the size of the environment table. You can do this at the command line (which is temporary) or by editing a configuration file to make a permanent change (which requires restarting your computer).

To temporarily increase environment space, enter the following at the DOS prompt:

```
command /e:8192
```

To permanently increase environment space, add the following line to your CONFIG.SYS file and then restart your system:

```
shell=command.com /e:8192 /p
```

Once you've increased the size of the environment table, try running Xeena again.

Unix

- Set your JAVA_HOME environment variable to be the absolute pathname of the JDK's top directory. For example, you can set JAVA_HOME in a shell by the following commands (when JDK is installed in directory /usr/jdk1.1.6):
JAVA_HOME=/usr/jdk1.1.6
export JAVA_HOME
- In the Xeena directory, run Xeena by executing:
 - **sample_dtd.sh**
Xeena will start using the sample address book DTD.
 - **sample_xsd.sh**
Xeena will start using the sample address book XML Schema.
 - **editschema.sh**
Xeena will start using the W3C XML schema dtd allowing you to create/edit/view XML schema documents. Open addressbook.xsd to view the Addressbook sample schema. Use the menu option File->import->DTD in order to import any DTD document into an XML schema document. The XML Schema editor uses the 17 December 1999 Working Drafts for [Structures](#) and [Datatypes](#) (the version supported by the XML parser Xeena uses). The files structures.dtd datatypes.dtd and structures.profile reside in the samples directory.
 - **configdtd.sh**
Xeena will start in customizing mode allowing you to create a profile file for the address book DTD (we already created such a file which is opened automatically).
 - **configxsd.sh**
Xeena will start in customizing mode allowing you to create a profile file for the address book XML Schema (we already created such a file which is opened automatically).
 - **configxeena.sh**
Xeena will start in customizing mode allowing you to create a profile file for Xeena itself. Xeena starts with an opened profile file (located in the same directory as the addressbook DTD). This was achieved by using the flag "-xml".

Known problems

- Due to some bugs in the Swing implementation of the HTMLEditorKit, the help panel may be unreliable. You can choose the "Xeena help" item on the menu (windows users) or load the file /help/index.html from the installation into your browser (others).
- Xeena will not currently prevent you from saving an XML document in an encoding that does not support some of the characters it contains. When this is done, you will see question marks replacing the problematic characters when viewing the created file using the "View XML source" option.
- It is currently impossible to use DTD/XML-Schema's which reside in directories whose name includes a space.
- Due to a parser bug, opening an XML file whose DOCTYPE refers to a nonexistent external ID may cause undefined results - files open with no elements, and Xeena's operation may be affected thereafter, even for

correct files.

- Due to a bug in JFileChooser when starting xeena you may get:
java.lang.NullPointerException
 at javax.swing.plaf.basic.BasicDirectoryModel\$DoChangeContents.run(Comp)
 at javax.swing.SystemEventQueueUtilities.processRunnableEvent(Compiled)
 at javax.swing.SystemEventQueueUtilities.access\$1(Compiled Code)
 at javax.swing.SystemEventQueueUtilities\$RunnableTarget.processEvent(Co)
 at java.awt.Component.dispatchEventImpl(Compiled Code)
 at java.awt.Component.dispatchEvent(Compiled Code)
 at java.awt.EventDispatchThread.run(Compiled Code)

Ignore it and wait for xeena to start.

- The XML parser (xerces 1.0.3) does not accept the datatype element with a source attribute that equals to NMTOKEN, use string instead. Here is an example:

```
<element name="job-info">
  <type content="empty">
    <attribute default="no" name="is-manager">
      <datatype source="NMTOKEN">
        <enumeration value="yes"/>
        <enumeration value="no"/>
      </datatype>
    </attribute>
  </type>
</element>
```

Limitations

- All the limitations of the XML parser xerces 1.0.3 that are related to XML Schema apply in Xeena too (please read the documentation provided with xerces 1.0.3 for details).

Acknowledgements

Thanks to Don Brutzman and his students for testing several intermediate versions of Xeena.

Have fun using Xeena!

Shlomit Shachor-Ifergan & Sigalit Ur

XEENA

XML Editing Environment, Naturally in Java

(C) 1999-2000 IBM Corporation. All rights reserved.

XEENA is brought to you by the IBM Haifa Research Lab, Israel.

Please send comments to helpmap@il.ibm.com

and check our web site on alphaWorks

<http://www.alphaWorks.ibm.com/tech/xeena>

Contents

- 1.0 [Overview](#)
- 2.0 [User Interface Aids](#)
 - 2.1 [Actions Toolbar](#)
 - 2.2 [Toolbar Palette](#)
 - 2.3 [Sidebar Palette](#)
 - 2.4 [Work Area](#)
 - 2.4.1 [Building the XML File Tree](#)
 - 2.5 [Edit area](#)
 - 2.6 [Message Area](#)
- 3.0 [Encoding](#)
- 4.0 [Sample Editing Session](#)
- 5.0 [Configuring Xeena](#)
- 6.0 [Using Xeena as a BML editor](#)

1.0 Overview

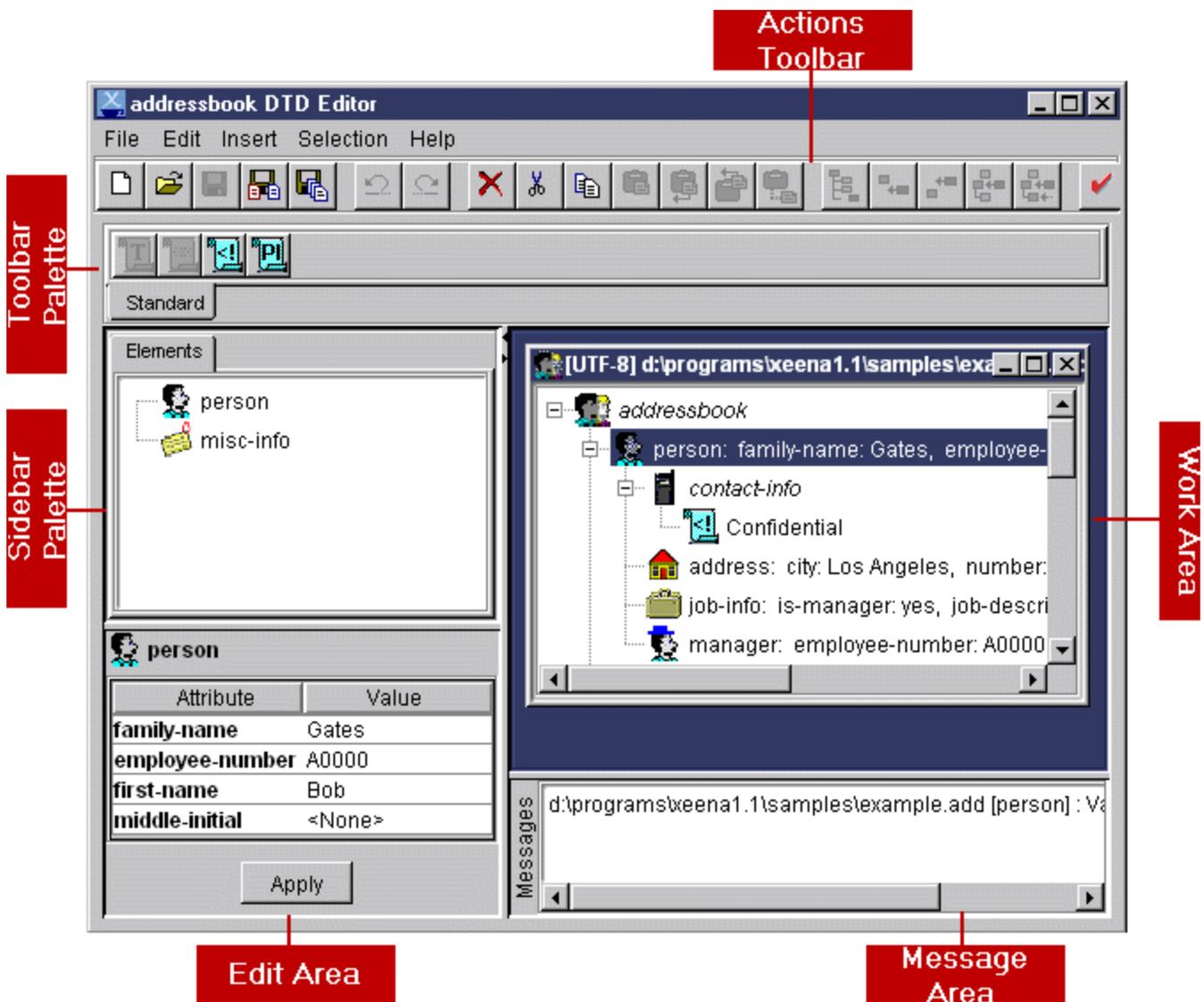
This document describes the Xeena XML editor. A [step-by-step tutorial](#) is also available.

The Xeena window includes the following items:

- Menu bar - It contains the following menus: File, Edit, Insert, Selection, Tools and Help. All the items in the Menu bar have corresponding buttons on the Actions toolbar, except for Export, Import and Close under

the File menu.

- [Actions toolbar](#) - It contains graphical buttons that enable you to choose actions appropriate for a selected node. Enabled action items on the Actions toolbar (i.e. Actions which are possible for the selected element) are highlighted.
- [Toolbar palette](#) - It contains graphical buttons of elements that can be selected. Once an element is selected, an available action for it is selected from the Actions toolbar.
- [Sidebar palette](#) - It contains a list of elements that can be selected. Once an element is selected, an available action for it is selected from the Actions toolbar.
- [Work area](#) - displays opened XML files, and relevant dialog boxes.
- [Edit Area](#) - an area where you can edit the data associated with the selected node in the active document. For example, for an XML Element it displays a table for editing attributes, for a PI it displays text areas where you can type the PI's target and data etc.
- [Messages area](#) - displays messages regarding the current process, for example, error messages.



2.0 User Interface Aids

Xeena includes several aids to make the user interface more user-friendly:

- When holding the mouse cursor over a button or palette component, a 'hover' area displays the name of the button. When hovering over an element button or palette component, the 'hover' area will show important information about the element. Hover help is displayed even for disabled buttons.
- When hovering over an attribute name in the attributes window, you'll see information about the attribute.
- Invalid document nodes (e.g. for an XML Element due to incomplete structure or invalid attributes values) are underlined in red. Validity is checked immediately after opening a file and during editing. This

mechanism provides immediate feedback on the validity of the edited document.

- The Palette and Message areas contain sashes that enable you to resize them. These sashes are located in the top left hand corner of the grid.

2.1 Actions Toolbar

The Actions toolbar contains buttons that provide quick access to frequently used editing commands. You can click an Actions button instead of selecting the appropriate item from the Menu toolbar. The availability of the button commands varies depending on the types of actions allowed for the currently selected node. An Action button is active if it is highlighted and gray if it is not available. This automatic enable/disable feature guides the user in making choices. The Actions toolbar contains the following buttons:

Icon	Name	Explanation
	New	Creates a new file.
	Open	Opens an existing file.
	Open URL	Open a file given its URL.
	Save	Saves a modified file.
	Save As	Saves a newly created file.
	Save All	Saves all open files.
	Undo	Undoes an action (up to the last 10 actions).
	Redo	Redoes an undone action.
	Delete	Deletes a selected element together with any children.
	Cut	Cuts a selected element together with any children and places it in the buffer.
	Copy	Copies a selected element together with any children and places it in the buffer.
		Pastes the contents of the buffer, replacing the

	Paste	current element.
	Paste After	Pastes the contents of the buffer after the selected element.
	Paste Before	Pastes the contents of the buffer before the selected element.
	Paste as Child	Pastes the contents of the buffer as a child of the selected element.
	Insert as Child	Inserts a element from the Palette or the Elements toolbar, as a child of the selected element.
	Insert After	Inserts a element from the Palette or the Elements toolbar after the selected element.
	Insert Before	Inserts a element selected from the Palette or the Elements bar before the selected element.
	Replace keep target children	Replaces the selected element with an element from the Palette or the Elements toolbar but keeps the original target children.
	Replace remove target children	Replaces the selected element with an element from the Palette or the Elements toolbar and removes the original target children.
	Validate	Validates the selected element.
	Expand	Expand all selected element's descendants.
	Collapse	Collapse all selected element's descendants.
	View XML src	Shows the XML source of the selected element.
	Print XML src	Prints the XML source of the selected element.
	Print XML tree	Prints the tree representation of the selected element.
	Process XSL	Invoke XSL processor on the active document.
	About	Provides details of the version of the product.

	Help Topics	Opens the help file.
---	-------------	----------------------

2.2 Toolbar Palette

The Toolbar palette provides options for building XML files according to the DTD/XML-Schema. As for the Actions, only those elements that are available for the particular definition are enabled. The Toolbar palette is divided into two sections; the top part contains alternative toolbar types and the bottom part contains the elements for each toolbar type.

There are three ways to include an element from the Toolbar palette into the XML file. Select the icon of the required element and then do one of the following:

- Choose a menu item from the popup menu.
- Choose a menu item from the Edit or Insert menu.
- Choose an action from the Actions toolbar.

2.3 Sidebar Palette

The Sidebar Palette contains ready-to-use elements for creating an XML file. There are two palette modes: in "all visible" mode, all the elements are visible at all times, and only the relevant ones (according to the DTD/XML-Schema) are enabled. In the second mode (the default mode), only the relevant elements are visible on the palette. See [Customizing Xeena](#) for details on how to set the desired mode.

There are four ways to include a Palette Element into the XML file. Select the required element and then do one of the following:

- Choose a menu item from the popup menu.
- Choose a menu item from the Edit or Insert menu.
- Choose an action from the Actions toolbar.
- Double-click the element to insert it as a child of the selected element in the tree (option available only if valid).

2.4 Work Area

The work area displays the files currently being edited, in a familiar MDI

fashion. Each XML file is shown as a tree, with the document root as the tree root. You can add, delete, cut, copy and paste elements into the tree to create your XML file. Invalid elements are underlined in red. Pressing the "Validate" Action displays the invalidity reasons in the message area.

2.4.1 Building the XML File Tree

The rules for building the tree, i.e. what elements are legal for each part of the tree, are dictated by the DTD/XML-Schema. Only elements (on the Sidebar and Toolbar palettes) that are allowed to be inserted at the current stage are available. Moreover, for the selected statement, the available actions on the Action toolbar are enabled. By disabling illegal insertions, the user interface does not allow to build an illegal tree.

2.5 Edit Area

An editable item appears in normal font in the document (non editable items appear in italic font). Selecting an editable item displays, depending on the item's type (Element, Comment, Text, CDATA section, Processing Instruction), an appropriate UI where it can be edited. For example, attributes in a table are displayed for Elements. Press the "Apply" button to apply your changes to the item, or press the "Undo" toolbar Action to restore the previous state.

Invalid content in the edit area (e.g. an invalid attribute value) is colored red; hovering over an invalid value will show the reason the value is invalid.

2.6 Message Area

When you attempt to save a file that contains errors, error messages appear in the Message area of the main window. Errors may occur as a result of incomplete tree structure, invalid attributes values or invalid characters in standard elements (PI, Comment, CDATASection). Double-clicking a message will highlight the problematic element in the file you're attempting to save.

3.0 Encoding

By default, new XML documents are created using the UTF-8 encoding. You can change the document encoding by selecting the first element in the

document, and setting its "encoding" attribute. You will be asked to select from a list of possible encodings. You should note that most of these are not widely supported.

It is also possible to define a default encoding for new documents for a specific DTD/XML-Schema by setting the "encoding" attribute of the "defaults" element in the DTD/XML-Schema profile (see [instructions for Xeena configuration](#)).

Xeena will not currently prevent you from saving an XML document in an encoding that does not support some of the characters it contains. When this is done, you will see question marks replacing the problematic characters when viewing the created file using the "View XML source" option.

4.0 Sample Editing Session

This section will lead you through a step-by-step process to build a sample file in Xeena. You will build a sample addressbook file, which will include one person, and save the file.

1. Opening a new file:

On the File menu, select New or click the New icon on the Actions toolbar. A new file window will be opened in the work area, which contains an 'addressbook' element as the document root. Note that the root is selected and underlined in red (has incomplete content since it must have at least one 'person' child).

2. Editing the file:

The Sidebar palette shows the "person" element, as it is the only element that can be inserted into the 'addressbook' element (according to the DTD). Right-mouse-click on the 'person' item in the Sidebar palette, and select the "Insert as child" option. The 'person' element will be inserted into the tree, and selected. The 'person' element is underlined in red to show that it is invalid due to invalid attribute values and incomplete content (the DTD specifies that the attributes are required and that the "person" element must have several children). You will now see in the edit area a table containing the selected 'person' attributes. Invalid values are colored red. Enter values into the table until all entries are valid. Note that the 'person' element is still underlined in red since it still invalid (has

an incomplete content).

3. Saving the file:

Try saving the file: on the File menu, select Save or click the Save icon on the Actions toolbar. A File chooser dialog is displayed. Choose a file name and press the 'Save' button. The file has been saved, but is invalid therefore you will get a beep, and an error message in the Messages area, stating "Incomplete content - at least one child is missing".

Double-click the message, and the file window will expand, while the 'person' element who is missing children is selected. Add the missing children. When you are done the 'person' element should no longer be underlined in red indicating that it is valid.

5.0 Customizing Xeena (optional)

Xeena can be configured in two levels: *general*, which affects all invocations of Xeena, and *DTD/XML-Schema-specific*, which is specific to a particular DTD/XML-Schema. When configuring at the general level, you can change the icons used in the actions toolbar, as well as the color scheme Xeena will use. When configuring at the DTD/XML-Schema-specific level, you can specify the icons used for the DTD/XML-Schema elements, which tooltips appear for elements and attributes, how attributes are edited, if the sidebar/toolbar palettes should be displayed and what elements the displayed palettes contain.

Xeena uses configuration files named *.profile - xeena.profile for general Xeena configuration, and <DTD/XML-Schema-name>.profile for DTD/XML-Schema-specific profiling. It looks for these files in the USER_HOME directory, and if it does not find them, in the directory in which the DTD/XML-Schema resides.

To configure Xeena at the general level, run Xeena with the following command (see [Usage](#) Section in the README file):

```
xeena -cfg
```

If a xeena.profile file already exists in the USER_HOME it is opened automatically otherwise you should open a new file. In case of a new file you will be shown a default configuration for Xeena. You may proceed to change this configuration as needed, then save the configuration file and exit.

Here's an explanation of the general configuration file tree structure:

-  **Xeena profile** - root. Set the 'item-editor' attribute to specify if item editors are displayed via a dialog or in the Edit Area.
 -  **Color scheme of Xeena desktop** - edit attributes to change colors
 -  **Icons of Xeena desktop** - edit attributes to change icons

To configure Xeena for a particular DTD/XML-Schema, invoke Xeena with the following command (see [Usage](#) Section in the README file):

```
xeena -cfg -dtd <dtd-file> [ -root <element name>,<element name>,.... ]
```

or

```
xeena -cfg -xsd <xml-schema-file> [ -root <element name>,<element name>,.... ]
```

If a <DTD/XML-Schema-name>.profile file already exists either in USER_HOME or in the DTD directory it is opened automatically otherwise open an new file. In case of a new file you will be shown a default configuration for the DTD you specified. You may proceed to change this configuration as needed, then save the configuration file and exit. The resulting configuration file will be used in future invocations of Xeena for the DTD/XML-Schema you specified.

The -root qualifier is optional; if unspecified, any element can serve as a root.

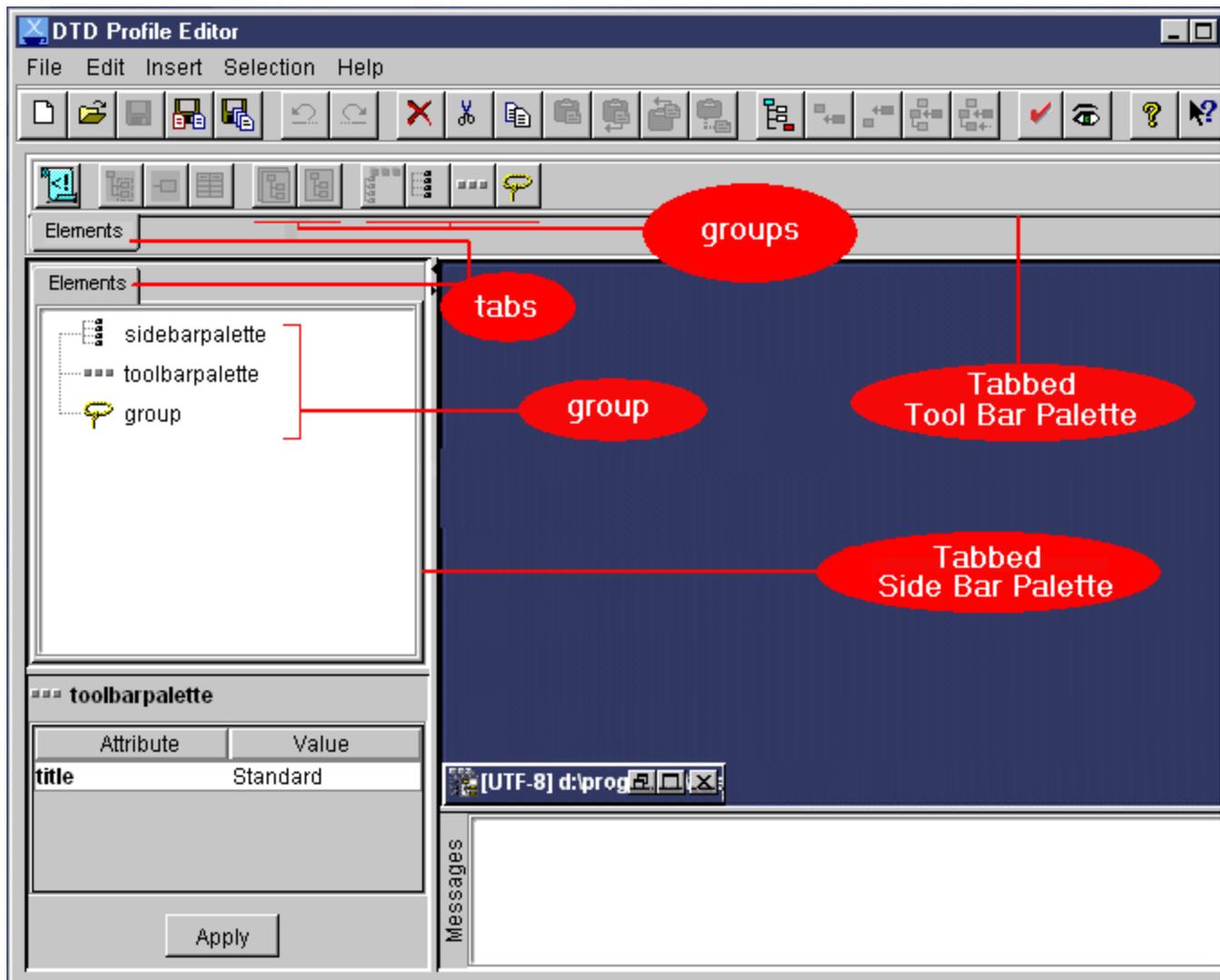
Here's an explanation of the DTD/XML-Schema-specific configuration file tree structure:

-  **DTDProfile** - root. Set the 'title' attribute to specify the title of the editor. Set the base-path attribute to specify a path to server as a base for relative file names in other attributes (e.g. icon files).
 -  **defaults** - defaults for document. Set the systemLiteral attribute to specify the default system external ID in new files. Set the encoding attribute to specify the default encoding for new documents. set the pubidLiteral attribute to specify the default public identifier for new documents.

-  **elements** - list of elements in DTD/XML-Schema
 -  **element** - maps to one element in the DTD/XML-Schema. Set the "tooltip" attribute to specify the tooltip for the element. Set the icon attribute to specify the icon to be used for that element.
The order of the element's *attributes* children dictates their order in the attributes table (displayed in the Edit Area) and in the description of the element (displayed in the document-tree).
 -  **attribute** - maps to an attribute of the element. Set the "tooltip" attribute to specify the tooltip for the attribute. Set the "long-text-editor" attribute to enable editing a multi-line value for the attribute (relevant only when the attribute is of type CDATA).
-  **documents** - types of documents that you want to customize.
 -  **document** - definition of a document. Set the "template" attribute to specify a file which should be used as a base when opening a new document; set the "suffix" attribute to specify the default file suffix for documents of this type; set the "root" element to specify the root of this document (i.e the type).
-  **palettes** - palettes to appear on the Xeena interface. The children **toolbarpalette** and **sidebarpalette** are optional. If not present, the missing palette will not be displayed. You can use this option to configure Xeena to display only a sidebar palette or a toolbar palette (both are displayed by default).
 - **toolbarpalette** - contents of toolbar palette. Each toolbarpalette defines a single toolbar palette. You may define more than one toolbarpalette, in this case all the tool bar palettes will be organized in a single tabbed tool-bar where each tab represents a single toolbar.
 -  **group** - a named group of elements. Set the "elements" attribute to the list of elements in this group. Each 'toolbarpalette' may contain more than one group. In this case elements of one group are separated from elements of the other groups by a separator.
 -  **sidebarpalette** - contents of sidebar palette. Each sidepalette defines a single sidebar palette. You may define more than one sidebarpalette, in this case all the sidebar palettes will be

organized in a single tabbed side-bar where each tab represents a single sidebar.

-  **group** - a named group of elements. Set the "elements" attribute to the list of elements in this group.
-  **importers** - contents of Import menu item.
 -  **importer** - an importer from an external format into an XML format. See the [FAQ](#) for instructions on how to implement importers.
-  **exporters** - contents of Export menu item.
 -  **exporter** - an exporter from an XML format into an external format. See the [FAQ](#) for instructions on how to implement exporters.
-  **tools** - contents of the Tools menu.
 -  **tool** - a tool to be invoked on a document. See the [FAQ](#) for instructions on how to implement tools.



6.0 Using Xeena as a BML editor (optional)

To demonstrate Xeena's hooks, a Xeena profile customized to edit BML (Bean Markup Language) files has been provided (see the [BML home on alphaWorks](#) for a description). To use this sample, follow the following steps:

- Download and install BML according to the [instructions on the alphaWorks site](#) (Xeena has been tested with BML 2.4 only).
- Define an environment variable BML_HOME that points to the root directory of the BML installation (e.g. c:\bml-2.4).
- Execute the batch file bml.bat (for Windows) or the shell file bml.sh (for Unix).

You can now follow this step-by-step demonstration of the customized editor's capabilities:

- Examine the sample juggler.bml file that has opened when you invoked the batch file. It describes a simple juggler application, which is supplied as a demo in the BML distribution. The application contains a juggler bean, along with 2 buttons which start and stop animation.
- Invoke Tools->Play BML, and see the application in action.
- Add a label component to the bottom of the application panel, by following the following steps:

Create a label bean, labeled "IBM juggler":

- Invoke File->Import->BML, and enter the bean name "java.awt.Label" in the ensuing dialog box. Press OK.
- Import the bean into a new document. You now have a document which describes a Label bean.
- Set the label's text by selecting the property element whose name is text, then setting its value to "IBM juggler".

Insert the label bean you created into the juggler bean, specifying that it should appear at the bottom of the window:

- Copy the resulting instantiated bean by selecting the "bean" element, and invoking the "Copy" popup menu item.
- Select the juggler.bml document.
- Add an "add" element after the 3 "property" elements at the beginning of the document.
- Paste the "label" bean you copied from the imported document as a child of the new "add" element.
- Add a new "string" element after the "bean" element, and set its value to "South" (the main panel of the juggler bean is in BorderLayout, and the new bean you're inserting will reside at the bottom of that panel).

Test the results:

- The resulting document should now be identical to the document named "labeled.bml" in the samples directory. Examine the resulting bean by invoking the BML player again.

- You can generate the program you've seen in action by using the File->Export->BML option.