INTEGRATION PAPER

XENAT

XML Editing New Authoring Tool

AT4LEX

**Purpose:**

The goal of this document is to outline the architecture and integration mechanisms for the use of Xenat (AT4LEX Editor). Aimed to technical and functional persons it is only an overview as more details can be found in technical documents.

Table of contents

[1. Xenat description 3](#_Toc463885365)

[2. Definitions: TEXT and ELEMENT 4](#_Toc463885366)

[3. Current XENAT integration: 5](#_Toc463885367)

[4. Future XENAT integration: 6](#_Toc463885368)

[4.1. Integration strategies: 6](#_Toc463885369)

[4.2. Main principles: Plugins, Xmlrules, Profiles 6](#_Toc463885370)

[4.3. Plugins. 6](#_Toc463885371)

[4.4. XMLrules 6](#_Toc463885372)

[4.5. Profiles (configuration groups) 7](#_Toc463885373)

[5. Recommended integration : URL TO XENAT 10](#_Toc463885374)

[6. Embedded: 11](#_Toc463885375)

[6.1. Embedded integration advantages/inconvenient: 11](#_Toc463885376)

[6.2. Integration schema 11](#_Toc463885377)

[6.3. Html5 example 13](#_Toc463885379)

[1. ANNEX FEATURES 14](#_Toc463885380)

[2. Annex - Document control 15](#_Toc463885381)

[2.1. Change history 15](#_Toc463885382)

[2.2. Glossary 15](#_Toc463885383)

# Xenat description

In this first chapter, you will find a short description of Xenat. It’s not the purpose of this chapter to fully describe the editor but only to outline some features that are important for the integration with other systems.

Xenat is a web editor that can edit content (text) and add[[1]](#footnote-1) new elements in a controlled way.

Xenat manage XML (XML4EP) documents. XML4EP is dividing clearly the text in different elements (e.g paragraphs, subparagraphs, citations, recitals....). The users via Xenat will be able to change the elements and/or the text of the elements. Different rules for different document types assure the consistency of documents[[2]](#footnote-2).

Xenat helps the users to apply the existing EP rules during the document editing process.

Xenat generates an XML[[3]](#footnote-3) in a transparent way for the user.

Xenat includes versioning control[[4]](#footnote-4):

* Auto check-out.
* Multiple users can change different parts at the same time.
* Only one user can change 1 part at the same time.
* Auto check-in after a period of inactivity[[5]](#footnote-5).

AT4LEX Dashboard or an external application can start Xenat via an URL: 4 parameters must be defined in this URL:

1. Format: the formatting rules[[6]](#footnote-6) used to show the document on the screen.
2. Content: the UBI[[7]](#footnote-7) of the document to be opened.
3. Mode and user: Read only (RO) or read/write mode.
4. Signature: To check against a public key all the other parameters.

Below an example of parameters used in the url

1: .......?style=styles\_for\_PR\_INI

2: &UBI=eu.europa.europarl-DIN1-2016-0000005186\_01.00-en-01.00\_text-xml...........

3: &user=jcerro&mode=RW

4: &signature=302C021215F21DDD...........................................

# Definitions: TEXT and ELEMENT

Xenat is an XML editor presenting the text by elements. An element is a part of text. Examples of elements are citations, recitals, paragraphs, titles, subparagraphs, articles....

Xenat is facilitating[[8]](#footnote-8), via defined rules, the element management by the user: create elements, delete, move, merge........

Each element could have an associated text. Xenat is helping the user to manage (edit) the content (text) of these elements.

In summary, with Xenat a user is able to manage the XML elements and/or the content of the elements.

# Current XENAT integration:

Today XENAT is developed using GWT. 2 ways of integration with XENAT exist:

1- URL, **recommended** Opening XENAT in the Browser. The user will have access to all of the XENAT features. The calling application should define the Format and the UBI. The integration effort is minimum and all the features are included.[[9]](#footnote-9) Including the versioning management (check-in/out).

2- Embedded: Import the GWT code and customize. The application itself must be based on GWT.

Inconvenient:

The XENAT team does not support any Compatibility problems that may arise in the GWT application.

It’s recommended that the application uses the same GWT and Spring version.

Event processing collisions and maybe widgets conflicts are possible between the application and XENAT.

Future XENAT versions will be difficult to integrate.

# Future XENAT integration:

We are implementing the next XENAT version avoiding the utilisation of any UI framework. We want to facilitate the integration in any application. We are using JAVA, HTML5, CSS and Javascript.

## Integration strategies:

1- URL **recommended** See 2.1. The application calls Xenat. The editor will be opened in another tab or browser window. Full editor features including version management[[10]](#footnote-10) and persistence are supported.

2- Embedded: The application should import the XENAT javascript library xenat.min.js “Minified version” (<script src="js/xenat.min.js"></script>). The application should also import the JS libraries of any plugin that it uses.

## Main principles: Plugins, Xmlrules, Profiles

Xenat is a modular design based on plugins. These plugins are configurable via profiles. Xenat controls the XML structure via XMLrules. XMLrules and profiles are different per document category[[11]](#footnote-11).

## Plugins.

1. **Control module:** Must be always present, it is in charge of element management and text management. It communicates with the rest of the plugins.
2. **Outline plugin:** Shows a tree representation of the document. Facilitates the element management. Needs Control Module.
3. **Tool-bar plugin:** Shows the buttons for different actions. Needs Control Module.
4. **Info bar pug-in:** Shows information about the document, the user and the status. Needs Control Module.
5. **SmartText plug-in:** Start the SmartText engine to generate automatically text. Needs Control Module.
6. **Application space plug-in:** Shows information provided by the app in html format. Images must be embedded in the HTML code.

## XMLrules

The XMLrules depend on the XML4EP definitions of each document category. Example of document category are: Draft Report, Explanatory Statement, Motion for Resolution, AM document, AM, CRE...

The rules are defined during the XML4EP setup and analysis of each document category.

## Profiles (configuration groups)

The integrating application team together with AT4LEX (Xenat) team should define the configuration for each plugin per application and document category. These profiles are used to setup the **configuration** for each plugin and for each document category for the specific application.

In the case of AT4LEX these profiles are already defined. The profile definitions are stored in DB via scripts. No interface to change these profiles is available. Cchanges, if needed, are done “manually”.

The possible configuration per plugin is listed below. This is not a complete list and the list could evolve. The plugin configuration is specific to the application and can be different for each XML4EP document catetgory supported by the application.

**Control Module**: Show text and allow to edit. Element management. Mandatory control.

|  |  |
| --- | --- |
| Feature | Config property |
| Plugin position | Left, rigth, center, up, down |
| Show text. | StyleName[[12]](#footnote-12) ( or default style) |
| Change text. Element by element. | Enabled/Disabled |
| Add quick text. | Enabled/Disabled |
| Add quick footnotes. | Enabled/Disabled |
| Spell checker. | -----Always active------------ |
| Remove elements (via DEL shortcut). | Enabled/Disabled |
| Add proposed elements (flying menu). | Enabled/Disabled |
| Move elements (via arrows shortcut). | Enabled/Disabled |
| Italic, bold, underline. | Enabled/Disabled |
| Special characters. | Enabled/Disabled |
| Superscript, subscript. | Enabled/Disabled |
| Non-breaking space. | ----Always active----- |

**Outline pluging**: Show the outline tree.

|  |  |
| --- | --- |
| Feature | Config property |
| Plugin position | Left, rigth, center, up, down, none[[13]](#footnote-13) |
| Show tree | StyleName ( or default style) |
| Bydefaultthreeopen | Closed/1st level open/2nd level/full |
| Select element | Enabled/Disabled |
| Change element type | Enabled/Disabled |
| Move elements. | Enabled/Disabled |
| Delete element. | Enabled/Disabled |

**Tool bar plug-in:** show the buttons for other actions.

|  |  |
| --- | --- |
| Feature | Config property |
| Plugin position | Left, rigth, center, up, down,none |
| Show toolbar | StyleName ( or default style) |
| Print button | Enabled/Disabled |
| Save section button | Enabled/Disabled |
| Modify element button | Enabled/Disabled |
| New element button | Enabled/Disabled |
| Delete element button | Enabled/Disabled |
| Copy & paste element buttons | Enabled/Disabled |
| PasteXL button | Enabled/Disabled |
| Merge element buttons | Enabled/Disabled |
| Back/Forward buttons | Enabled/Disabled |
| Add Terms button | Enabled/Disabled |
| Add footnote button | Enabled/Disabled |

**Info bar plug-in:** Show information about the document, the user , the status, the ids.....

|  |  |
| --- | --- |
| Feature | Config property |
| Plugin position | Left, rigth, center, up, down,none |
| Show info bar | StyleName ( or default style) |
| Chararacter count | Enabled/Disabled |
| Charlimit warning | Enabled/Disabled |
| Charlimit | # of characters |
| Position: Windows dressing | Enabled/Disabled |
| Locking user | Enabled/Disabled |
| Document ids | Enabled/Disabled |

**Smart text plug-in:** Start the SmartText engine to generate automatically text via a Wizard.

|  |  |
| --- | --- |
| Feature | Config property |
| Plugin position | Left, rigth, center, up, down, none |
| Show Smarttext bar | StyleName ( or default style) |
| Create new SmartText (wizard) | Enabled/Disabled |
| Modify SmartText (wizard) | Enabled/Disabled |
| Delete SmartText | Enabled/Disabled |

**Application space plug-in:** Information provided by the application.

|  |  |
| --- | --- |
| Feature | Config property |
| Plugin position | Left, rigth, center, up, down, none |
| Show application info | StyleName ( or default style) |

# Recommended integration : URL TO XENAT

**The application calls Xenat that is opened in a new (or the same) browser tab.**

The parameters to pass in the URL are:

1. Profile name defining the configuration for each plugin.[[14]](#footnote-14)
2. Content: the UBI[[15]](#footnote-15) of the document to be opened.
3. Mode and user: Read only (RO) or read/write mode.
4. Signature: To check against a public key all the other parameters.

**Content and profile management, persistence:**

Xenat reads the XML defined by the UBI from PUREXML, reads the profile from DB and presents to the user the editor using the plugins and setting up the pre-defined configuration via the profile.

**Version management:**

Xenat also implements the auto save, auto check-out, auto check-in functions and version management.

**Section management:**

In case the UBI is composed of multiple document categories (e.g PR) or sections (e.g. RoP), Xenat presents them in different Tabs. Every time the user changes the tab, Xenat loads the appropriate XML and profile.

Sections could also be presented inside a table, in columns or rows.

**Application space plug-in:** The calling application should insert, via post, the html to be presented by this plugin. Small Images could be embedded in this html.

# Embedded:

The application should import the XENAT javascript library.

## Embedded integration advantages/inconvenient:

**Flexibility:** In order to offer maximum **flexibility** certain features are excluded from the embedded version. The integrating application sends an XML to the Xenat components and receives back the XML modified by the user.

What is **NOT** included in this library?

1. The library does not include **the persistence**; the editor plug-in will return an XML (LegalDocML). The integrating application[[16]](#footnote-16) should treat this XML. See 4.3 for more details.
2. The library also does not include the **version management features** using PUREXML, i.e. auto check-in, auto check-out, auto save, lock , unlock...
3. The library is not including the **section management via tabs**. The management of LegalDocML structure[[17]](#footnote-17) and its visualisation is up to the integrating application to implement.
4. The **profiling system** can only be used when the Xenat editor is opened using the calling URL method. The integrating application is fully free to manage and change the plugins configuration.

The application decides which Xenat plugins it will use and where to show them in its own pages as well as what to do with the XML.[[18]](#footnote-18)

## Integration schema

Plugins are by default listening and can also generate the standard events of Xenat. Example if a user selects a text element in the Text plugin, this element will be automatically selected in the Outline plugin.[[19]](#footnote-19)

## 

## Html5 example

Following is an example HTML document that could host Xenat as an embeddable library. Important elements for the integration are:

* Import of the base Xenat library (xenat.min.js)
* Import of the JS libraries (and relative CSS where needed) of the Xenat plugins that will be used by the application.
* DIVs with specific IDs (canvas1, toolbar1, tree1) which are where the main canvas (text area) and the plugins will be inserted.
* JS call to instantiate the xenat document and its plugins.



# ANNEX FEATURES

List of Xenat features:

* Text WYSIWYG
* Show outline
* Show user and documents ID.
* Tab management
* Save/Autosave, auto check-in (inactivity time out)
* Select an element
* Hover menu to edit , delete or add elements
* Modify text of an element
* Delete an element
* Add a new element
* Move an element
* Change element type
* Copy / paste text
* Copy / paste elements.
* Merge elements.
* Undo/Redo
* Paste XL
* Footnotes. View, Add, remove, delete, copy, move
* Quick foot notes
* Add terms (Docep Reference Lists)
* Add QuickText
* Spell check
* Comments, add ; edit; delete
* Diffing
* Track changes
* Unstructured texts: White bullet, point, dash, Numbers, Indent, Long Dash letter....
* Automatic renumbering
* Bold, Italic Underline, superscript, underscript.
* Non-breakable space
* Who is locking the document?
* SmartText
* Protect text elements: user cannot change these elements.
* Char count
* Char limit warning

# -Annex - Document control

## Change history

| **Version number[[20]](#footnote-20)** | **Status[[21]](#footnote-21)** | **Date** | **Initials** | **Summary of changes** |
| --- | --- | --- | --- | --- |
| 0.1 | Draft | 02/10/2016 | JCE | Creation |
| 0.2 | Draft | 04/10/2016 | GAR | Review |
| 0.3 | Draft | 07/10/2016 | JCE,GAR,AK | Full review; plugins and profiles |
| 0.4 | Draft | 10/10/2016 | JCE | Diagram |
| 1.0 | Final | 13/10/2016 | JCE,GAR,AK | Full review |

## Glossary

| **Abbreviation** | **Description** |
| --- | --- |
| XML4EP | XML standard based in Akomantoso. This standard is managed by the eParliament team. |
|  |  |

1. Add, delete, move and change. [↑](#footnote-ref-1)
2. See annex I for a full feature list. [↑](#footnote-ref-2)
3. XML4EP standard. XML facilitate the automation of text management during the legislative process: Verification, Translation, Consolidation, Adoption, Publication.... [↑](#footnote-ref-3)
4. Via an integration with PUREXML. [↑](#footnote-ref-4)
5. Time out to be defined by document type. Electricity cuts and similar events are coped with this feature. [↑](#footnote-ref-5)
6. This is another XML file defining the formatting of each XML element. [↑](#footnote-ref-6)
7. Ubi= «PUREXML» single id. [↑](#footnote-ref-7)
8. But also to control. [↑](#footnote-ref-8)
9. Customizations or RFC could be requested to the XENAT team. [↑](#footnote-ref-9)
10. Check-in check-out, lock, multiple users. Via PURE-XML. [↑](#footnote-ref-10)
11. XML4EP definition. A document category can contain other categories. PR=Motion+Explanatory Stat. Only 1 XML is showed at the same time. [↑](#footnote-ref-11)
12. Defined togheter with Xenat team: CSS [↑](#footnote-ref-12)
13. None: the plugin is not displayed. [↑](#footnote-ref-13)
14. For multiple section document is defining the first profile to be used. [↑](#footnote-ref-14)
15. Ubi= «PUREXML» single id. [↑](#footnote-ref-15)
16. Application that imports the Xenat library. [↑](#footnote-ref-16)
17. Each « Document » is represented in multiple XML. Xenat use tabs/tables to represent each XML. [↑](#footnote-ref-17)
18. Info application plugin is not needed in the embedded integration schema. [↑](#footnote-ref-18)
19. Integrating applications could also capture the events, but this will require more expertise. [↑](#footnote-ref-19)
20. [Naming convention: Procedure 'Program & Project naming convention'](http://www.standardsnet.ep.parl.union.eu/standards/cms/Accueil/preconisations/P_Methodologie) [↑](#footnote-ref-20)
21. Status: Draft, Final, Approved [↑](#footnote-ref-21)