



LIBROPLANTA XML EDITOR S1000D Edition

API Reference Guide

LP-LIBRO-API01-00

Issue No. 001, 2017-09-05



Copyright © 2014 Libroplanta LLC.

S1000D® is a registered trademark of ASD (AeroSpace Defense).

Computer software LIBROPLANTA XML EDITOR. S1000D Edition is registered in FIPS (Federal Institute of Industrial Property), registration number 2016617993, dated 20.08.2016.

Manufacturer:



Libroplanta LLC
Makarenko 22/28, Severodvinsk,
Arkhangelsk region, 164509, Russian
Federation

Applicable to:
All

End of data module

UNCLASSIFIED

LPEDIT-A-00-00-00-00A-001B-A

2017-09-05 Page 1



LIBROPLANTA XML EDITOR S1000D Edition

API Reference Guide

Table of contents

	Page
API Reference Guide.....	1
References.....	2
Description.....	2
1 Introduction.....	2
2 Script – Application level.....	3
3 Script – Window level.....	3
4 Script – Document level.....	3
5 API-interface description.....	3
5.1 XmlEditor (Namespace).....	3
5.1.1 Classes.....	3
5.1.1.1 ApiApplication.....	3
5.1.1.2 ApiDocument.....	7
5.1.1.3 ApiWindow.....	12
5.1.1.4 ContentEventArgs.....	17
5.1.1.5 DocumentClosingEventArgs.....	18
5.1.1.6 DocumentEventArgs.....	19
5.1.1.7 DocumentFileEventArgs.....	20
5.1.1.8 DocumentSavingEventArgs.....	20
5.1.1.9 Selection.....	21
5.1.2 Enums.....	24
5.1.2.1 CloseMode.....	24
5.2 XmlEditor.Model (Namespace).....	24
5.2.1 Classes.....	24
5.2.1.1 CommandLineOptions.....	24
5.2.1.2 ExtendedXNode.....	25
5.2.1.3 XTextPointer.....	26
5.2.1.4 XmlCursor.....	26
5.2.2 Enums.....	27
5.2.2.1 RPosition.....	27
6 Multimedia formats support.....	27
7 URL-schema support.....	28
8 Elements identifiers naming and assigning rules.....	29
9 User smart tags.....	29
10 User context rules.....	30
11 Document display styles.....	30



List of tables

1	References.....	2
---	-----------------	---

List of figures

1	Authorization Dialog before starting application.....	7
2	Fluent-controls creation demo.....	11
3	Regular controls creation demo.....	11
4	ApiDocument Examples.....	12
5	Creating backstage tab with the file list retrieved from Resource Resolution Service.....	17
6	Encasing fragment selection with verbatimText element.....	24
7	URL-schema registration.....	29

References

Table 1 References

Data module / Technical publication	Title
None	

Description

1 Introduction

Libroplanta S1000D XML Editor can be integrated with different CSDBs (Common Source Data Bases). API provide access to graphical user interface controls and events handling.

Libroplanta S1000D XML Editor API is based on Python (version 2.7.7) script files.

The application supports 3 script levels:

- Application ([Para 2](#))
- Window ([Para 3](#))
- Document ([Para 4](#)).

API classes, methods and properties are described in [Para 5](#).

The support of multimedia files is described in [Para 6](#).

The support of reserved URL-schema is described in [Para 7](#).

Element identifiers naming and assigning rules are described in [Para 8](#).

User smart tags are described in [Para 9](#).

User context rules are described in [Para 10](#).



Document display styles are described in [Para 11](#).

2 Script – Application level

The executing script path should be as follows:

- DISK:\Program files (x86)\Libroplanta\S1000D XML Editor\PythonScripts\Application.py

The example scripts with commented content are available in the file Application.py.

The Application level script handles the following events:

- opening document
- creating new document

3 Script – Window level

The executing script path should be as follows:

- DISK:\Program files (x86)\Libroplanta\S1000D XML Editor\PythonScripts\Window.py

The example scripts with commented content are available in the file Window.py.

4 Script – Document level

The executing script path should be as follows:

- DISK:\Program files (x86)\Libroplanta\S1000D XML Editor\DocTypes\[S1000D Issue]\PythonScripts\Document.py

The example scripts in commented form are available in the file Document.py.

The Document level script handles the following events:

- before closing document
- after closing document
- before saving document
- after saving document
- after changing selection of the document
- left-click on the document content.

5 API-interface description

5.1 XmlEditor (Namespace)

5.1.1 Classes

5.1.1.1 ApiApplication

Provides Libroplanta XML Editor application object and access to its properties and methods

Namespace: [XmlEditor](#)



Inheritance hierarchy:

[System.Object](#)

XmlEditor.ApiApplication

Syntax:

```
public class ApiApplication
```

Table 2 Properties




Name	Description
 Documents	ApiDocument collection <pre>public List<ApiDocument> Documents { get; }</pre>
 UserModel	User data storage model <pre>public object UserModel { get; set; }</pre>
 Windows	ApiWindow collection <pre>public ObservableCollection<ApiWindow> Windows { get; internal set; }</pre>

Table 3 Events



Name	Description
 DocumentCreated	Called after creating new document <pre>public static event DocumentCreatedHandler DocumentCreated { add; remove; }</pre>
 DocumentOpened	Called after opening document <pre>public static event DocumentOpenedHandler DocumentOpened { add; remove; }</pre>



Table 4 Methods





Name	Description
 <code>NewDocument (ApiWindow, String)</code>	<p>Opens new document in the specified window</p> <pre>public static void NewDocument(ApiWindow window, string templatePath)</pre> <p>Parameters</p> <p>window (ApiWindow) Window</p> <p>templatePath (String) Full path to the document template</p>
 <code>NewDocument (ApiWindow, String, String)</code>	<p>Opens new document in the specified window</p> <pre>public static void NewDocument(ApiWindow window, string version, string template)</pre> <p>Parameters</p> <p>window (ApiWindow) Window</p> <p>version (String) ASD S1000D Issue</p> <p>template (String) Document template (for example: description.xml)</p>
 <code>NewDocument (String)</code>	<p>Opens new document</p> <pre>public static void NewDocument(string templatePath)</pre> <p>Parameters</p> <p>templatePath (String) Full path to the document template</p>
 <code>NewDocument (String, String)</code>	<p>Opens new document</p> <pre>public static void NewDocument(string version, string template)</pre> <p>Parameters</p> <p>version (String) ASD S1000D Issue</p> <p>template (String) Document template (for example: description.xml)</p>



Table 4 Methods (Continued)

Name	Description
☰ OpenDocument (String)	Opening document public static void OpenDocument(string fileName) Parameters fileName (String) Full path to the file

5.1.1.1.1 Examples

5.1.1.1.1.1 Authorization Dialog before starting application

```
#Main entry point of starting application
def StartApplication():
    formAuth = AuthorizeForm()
    formAuth.ShowDialog() #Open dialog before starting XmlEditor
    pass

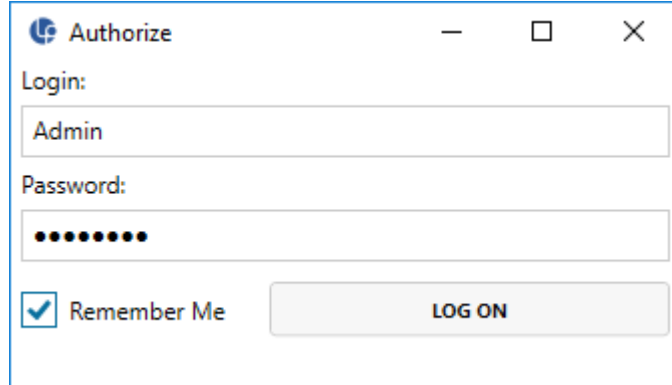
#Dialog creation
class AuthorizeForm(Window):
    def __init__(self):
        self.Title = "Authorize"
        self.Height = 200
        self.Width = 350
        self.AddControls()
        self.WindowStartupLocation = WindowStartupLocation.CenterScreen
        login = win.TextBox(Margin = Thickness(5), Height = 20)
        password = win.PasswordBox(Name = "password", Margin = Thickness(5), Height = 20)
    def AddControls(self):
        stackPanel = win.StackPanel()
        stackPanel.Children.Add(win.TextBlock(Text = "Login:", Margin = Thickness(5, 0, 0, 0)))
        stackPanel.Children.Add(self.login)
        stackPanel.Children.Add(win.TextBlock(Text = "Password:", Margin = Thickness(5, 0, 0, 0)))
        stackPanel.Children.Add(self.password)
        grid = win.Grid()
        btn = win.Button(Margin = Thickness(5), Content = "Log On", Width = 200, HorizontalAlignment = HorizontalAlignment.Right)
        btn.Click += self.ClickAuthorize
        grid.Children.Add(btn)
        grid.Children.Add(win.CheckBox(Margin = Thickness(5), VerticalAlignment = VerticalAlignment.Center, Content = "Remember Me", HorizontalAlignment = HorizontalAlignment.Left))
        stackPanel.Children.Add(grid)
        self.AddChild(stackPanel)
    def ClickAuthorize(self, s, e):
```




```

if self.login.Text == "admin" and self.password.Password == "password":
    self.Hide() #Hide the form and start XmlEditor
else:
    self.Close() #Close the application

```



ICN-LPEDIT-A-000000-A-LIBRO-E0601-A-001-01

Fig 1 Authorization Dialog before starting application

5.1.1.2 ApiDocument
 Provides access to the document model

Namespace: [XmlEditor](#)

Inheritance hierarchy:

[System.Object](#)
 XmlEditor.ApiDocument

Syntax:

```
public class ApiDocument
```

Table 5 Properties

Name	Description
FileName	Returns the full path (URI) to the opened document <pre>public string FileName { get;}</pre>
Ribbon	Provides access to the ribbon bar <pre>public Ribbon Ribbon { get;}</pre>
Selection	Provides access to the document fragment selection model <pre>public Selection Selection { get;}</pre>
TransactionManager	Provides access to the transaction manager (Undo - Redo) <pre>public UrManager TransactionManager { get;}</pre>
UserModel	User data storage model <pre>public object UserModel { get; set;}</pre>

Printed in Severodvinsk (Libroplanta LLC)

*Table 5 Properties (Continued)*



Name	Description
 Window	ApiWindow object <code>public ApiWindow Window { get;}</code>
 XDocument	Provides access to the XML document model <code>public XDocument XDocument { get;}</code>

Table 6 Events






Name	Description
 Content	Called when interacting with the document content. This event is subscribed in accordance with document style markup <code>public event ContentHandler Content { add; remove; }</code>
 DocumentClosed	Called after the document is closed <code>public event DocumentClosedHandler DocumentClosed { add; remove; }</code>
 DocumentClosing	Called before closing the document <code>public event DocumentClosingHandler DocumentClosing { add; remove; }</code>
 DocumentSaved	Called after the document is saved <code>public event DocumentSavedHandler DocumentSaved { add; remove; }</code>
 DocumentSaving	Called before saving the document <code>public event DocumentSavingHandler DocumentSaving { add; remove; }</code>



Table 7 Methods

Name	Description
☰ AddS1000DFile (String)	<p>Adds document or illustration to the support data</p> <p>public void AddS1000DFile(string file)</p> <p>Parameters</p> <p>file (String) Full path (URI) to the file</p>
☰ Close (CloseMode)	<p>Closes the document</p> <p>public void Close(CloseMode closeMode)</p> <p>Parameters</p> <p>closeMode Document close mode: (CloseMode)</p>
☰ CreateKeyBinding (Object, InputGesture)	<p>Binds shortcuts to methods</p> <p>public void CreateKeyBinding(object callback, InputGesture gesture)</p> <p>Parameters</p> <p>callback (Object) Target method</p> <p>gesture (InputGesture) Shortcuts</p>
☰ Save()	<p>Saves the document</p> <p>public void Save()</p>
☰ SaveAs (String)	<p>Saves the document with a new file name</p> <p>public void SaveAs(string fileName)</p> <p>Parameters</p> <p>fileName (String) Full path (URI) to the file</p>
☰ UpdateBrex()	<p>Updates loaded business rules</p> <p>public void UpdateBrex()</p>
☰ UpdateRepositorties()	<p>Updates loaded CIRs (Common Information Repositories) data</p> <p>public void UpdateRepositorties()</p>

5.1.1.2.1 Examples

5.1.1.2.1.1 Fluent-controls creation demo

```

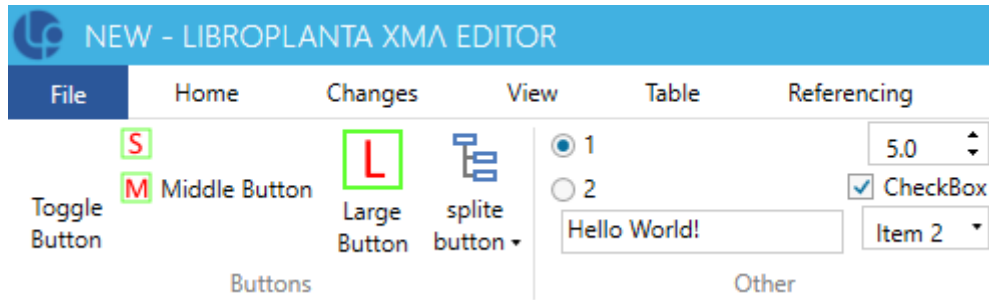
path = Environment.CurrentDirectory + "\\DocTypes\\S1000D_4.2.0\\PythonScripts\\"
#Fluent-controls
def ButtonsGroupBoxControls():
    buttonControls = Fluent.RibbonGroupBox(Header = "Buttons")

```

Printed in Severodvinsk (Libroplanta LLC)



```
largeButton = Fluent.Button(Header = "Large Button", LargetIcon = path + "icons\
\large.png")
largeButton.Size = Fluent.RibbonControlSize.Large
middleButton = Fluent.Button(Header = "Middle Button", Icon = path + "icons\
\medium.png")
middleButton.Size = Fluent.RibbonControlSize.Middle
smallButton = Fluent.Button(ToolTip = "Small Button", Icon = path + "icons\
\small.png")
smallButton.Click += lambda s, e:
MessageBox.Show(ApiDocument.Selection.ToString())
smallButton.Size = Fluent.RibbonControlSize.Small
toggleButton = Fluent.ToggleButton(Header = "Toggle Button", ToolTip
= "Notification...")
toggleButton.Click += lambda s, e: MessageBox.Show("Pressed: " +
str(toggleButton.IsChecked), "Button Status")
splitButton = Fluent.SplitButton(Header = "splite button", LargetIcon = path + "icons
\split.png")
gallery = Fluent.Gallery()
gallery.Items.Add(Fluent.Button(Header = "Button 1"))
gallery.Items.Add(Fluent.Button(Header = "Button 2"))
splitButton.Items.Add(gallery)
buttonControls.Items.Add(toggleButton)
buttonControls.Items.Add(smallButton)
buttonControls.Items.Add(middleButton)
buttonControls.Items.Add(largeButton)
buttonControls.Items.Add(splitButton)
return
buttonControls
def OthersGroupBoxControls():
otherControls = Fluent.RibbonGroupBox(Header = "Other")
checkBox = Fluent.CheckBox(Header = "CheckBox", IsChecked = True)
radioButton = Fluent.RadioButton(Header = "1", IsChecked = True)
radioButton2 = Fluent.RadioButton(Header = "2")
spinner = Fluent.Spinner(Maximum = 100, Minimum = 0, Value = 5)
textBox = Fluent.TextBox(Width = 150, Text = "Hello World!")
comboBox = Fluent.ComboBox(SelectedIndex = 1, IsReadOnly = True)
comboBox.Items.Add("Item 1")
comboBox.Items.Add("Item 2")
comboBox.Items.Add("Item 3")
comboBox.Items.Add("Item 4")
otherControls.Items.Add(radioButton)
otherControls.Items.Add(radioButton2)
otherControls.Items.Add(textBox)
otherControls.Items.Add(spinner)
otherControls.Items.Add(checkBox)
otherControls.Items.Add(comboBox)
return otherControls
```

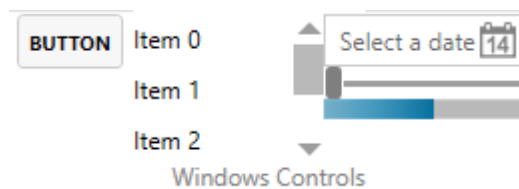


ICN-LPEDIT-A-000000-A-LIBRO-E0602-A-001-01

Fig 2 Fluent-controls creation demo

5.1.1.2.1.2 Regular controls creation demo

```
def WindowsGroupBoxControls():
    #https://msdn.microsoft.com/en-us/library/system.windows.controls(v=vs.110).aspx
    winControls = Fluent.RibbonGroupBox(Header = "Windows Controls")
    button = win.Button(Content="Button")
    listBox = win.ListBox(Width=100)
    for i in range(5):
        listBox.Items.Add("Item " + str(i))
    calendar = win.DatePicker()
    progressBar = win.ProgressBar(Maximum = 100, Value = 55)
    slider = win.Slider(Maximum = 100)
    winControls.Items.Add(button)
    winControls.Items.Add(listBox)
    winControls.Items.Add(calendar)
    winControls.Items.Add(slider)
    winControls.Items.Add(progressBar)
    return winControls
```



ICN-LPEDIT-A-000000-A-LIBRO-E0603-A-001-01

Fig 3 Regular controls creation demo

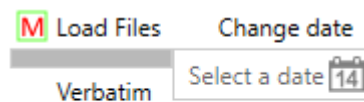
5.1.1.2.1.3 ApiDocument Examples

```
progressBar = win.ProgressBar()
calendar = win.DatePicker()
def ApiPanelSample():
    apiPanel = Fluent.RibbonGroupBox(Header = "API Panel Sample")
    #Loading files from folder
    btnLoader = Fluent.Button(Header = "Load Files", Icon = path + "icons\
    \medium.png")
    btnLoader.Size = Fluent.RibbonControlSize.Middle
    btnLoader.Click += ClickEventLoadFiles
    #Changing issue date to the selected date "calendar"
```



```
btnChangeDate = Fluent.Button(Header = "Change date")
btnChangeDate.Size = Fluent.RibbonControlSize.Middle
btnChangeDate.Click += ClickEventChangeDate
apiPanel.Items.Add(btnLoader)
apiPanel.Items.Add(progressBar)
apiPanel.Items.Add(btnVerbatim)
apiPanel.Items.Add(btnChangeDate)
apiPanel.Items.Add(calendar)

return apiPanel
def ClickEventChangeDate(s, e):
    try:
        #Start transaction
        ApiDocument.TransactionManager.StartTransaction("Date Changed")
        date = calendar.SelectedDate
        day = str(date.Day).zfill(2)
        month = str(date.Month).zfill(2)
        year = str(date.Year)
        doc = ApiDocument.XDocument.XPathSelectElement("//dmodule/
identAndStatusSection/dmAddress/dmAddressItems/issueDate")
        doc.Attribute("day").SetValue(day)
        doc.Attribute("month").SetValue(month)
        doc.Attribute("year").SetValue(year)
        #End transaction
        ApiDocument.TransactionManager.EndTransaction()
    except:
        #Rollback transaction in case of exception catching
        ApiDocument.TransactionManager.RollbackTransaction()
def ClickEventLoadFiles(s, e):
    progressBar.Maximum = Directory.GetFiles(path + "Files").Length
    for file in Directory.GetFiles(path + "Files"):
        ApiDocument.AddS1000DFile(file)
        progressBar.Value += 1
    MessageBox.Show("Done!")
    ApiDocument.UpdateRepositorties()
    ApiDocument.UpdateBrex()
```



API Panel Sample

ICN-LPEDIT-A-000000-A-LIBRO-E0605-A-001-01

Fig 4 ApiDocument Examples

5.1.1.3 ApiWindow Window object



Namespace: [XmlEditor](#)

Inheritance hierarchy:

[System.Object](#)

[XmlEditor.ApiWindow](#)

Syntax:

```
public class ApiWindow
```

Table 8 Constructors


Name	Description
 ApiWindow (Boolean)	Creates new application window <pre>public ApiWindow(bool startScreenEnabled)</pre> <p>Parameters</p> <p>startScreenEnabled Opens the start screen (Boolean)</p>

Table 9 Properties






Name	Description
 Backstage	Returns backstage-control <pre>public BackstageTabControl Backstage { get;}</pre>
 Document	Returns document opened in this window <pre>public ApiDocument Document { get; internal set;}</pre>
 UserModel	User data storage model <pre>public object UserModel { get; set;}</pre>

Table 10 Methods

Name	Description
 BringToFront()	Brings application window to front <pre>public void BringToFront()</pre>
 Close()	Closes the window <pre>public void Close()</pre>

*Table 10 Methods (Continued)*

Name	Description
NewDocument (String)	Creates new document in this window <code>public ApiDocument NewDocument(string templatePath)</code> Parameters templatePath (String) Full path (URI) to the document template
NewDocument (String, String)	Creates new document in this window <code>public ApiDocument NewDocument(string version, string template)</code> Parameters version (String) ASD S1000D Issue template (String) Document template (for example: description.xml)
OpenDocument (CommandLineOptions, CreatedOrOpened)	Opens document in this window <code>public ApiDocument OpenDocument(CommandLineOptions options, CreatedOrOpened createdOrOpened)</code> Parameters options (CommandLi-Command line parameters neOptions) createdOrOpened Document opening mode (CreatedOrOpened)
OpenDocument (String, CreatedOrOpened)	Opens or creates document in this window <code>public ApiDocument OpenDocument(string fileName, CreatedOrOpened createdOrOpened)</code> Parameters fileName (String) Full path (URI) to the document createdOrOpened Document opening mode (CreatedOrOpened)

5.1.1.3.1 Examples

5.1.1.3.1.1 Creating backstage tab with the file list retrieved from Resource Resolution Service

#Retrieving files from Resource Resolution Service

`def request():`

`wc = WebClient()`

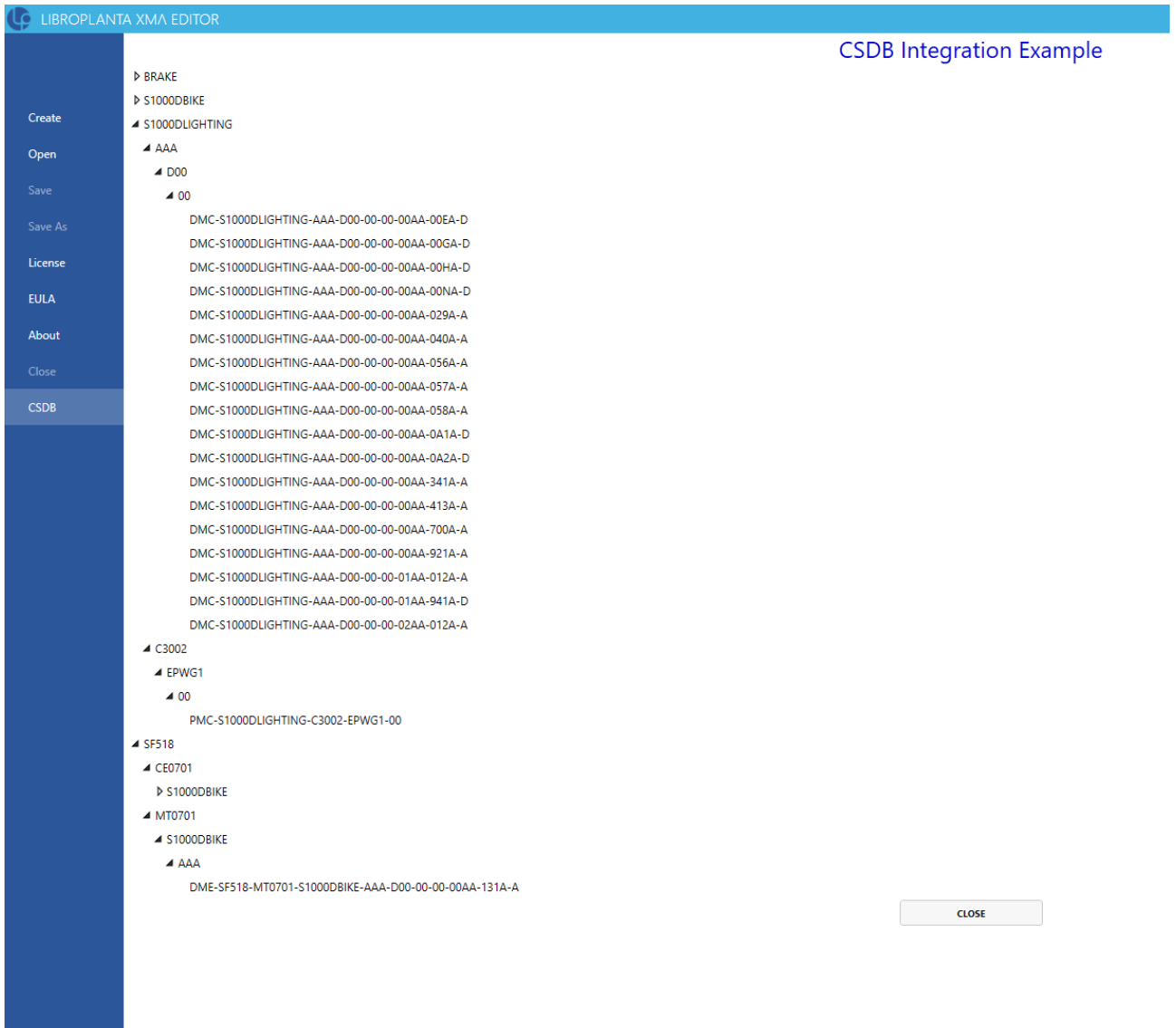
`wc.Encoding = Encoding.UTF8`



```
result = wc.DownloadString("http://resolver.libroplanta.com/ResolverA?
allurcs&lang=en")
doc = XDocument.Parse(result)
rdf = XNamespace.Get("http://www.w3.org/1999/02/22-rdf-syntax-ns#")
dc = XNamespace.Get("http://www.purl.org/dc/elements/1.1/")
files = doc.Root.Element(rdf + "RDF").Elements(rdf + "Description")
arr = []
for i in files:
    if not(i.Element(dc + "title") or i.Element(dc + "language")) : continue
    arr.append([i.Attribute("urn").Value, i.Attribute("about").Value])
return arr
#Hierarchical data model
class Item(object):
    def __init__(self, Name, Uri):
        self.Name = Name
        self.Uri = Uri
        self.Children = List[Item]()
#Main entry point of opening window
def LoadWindow():
    groupList = []
    items = List[Item]()
    for i in request():
        result = re.search(r'URN:S1000D:([A-Z]*)-(?P<name>[0-9A-Z]*)-(?
P<firstValue>[0-9A-Z]*)-(?P<twoValue>[0-9A-Z]*)-(?P<threeValue>[0-9A-Z]*)', str(i[0]
[0]))
        groupList.append([str(result.group('name')), str(result.group('firstValue')),
str(result.group('twoValue')), str(result.group('threeValue')), str(i[0]
[0].replace("URN:S1000D:", "")), str(i[1][0])])
        sortedGroupList = sorted(groupList)
#Tree view creation
for _layer1 in groupby(sortedGroupList, key = lambda x: x[0]):
    level1 = Item(_layer1[0], None)
    items.Add(level1)
    for _layer2 in groupby(_layer1[1], key = lambda x: x[1]):
        level2 = Item(_layer2[0], None)
        level1.Children.Add(level2)
        for _layer3 in groupby(_layer2[1], key = lambda x:x[2]):
            level3 = Item(_layer3[0], None)
            level2.Children.Add(level3)
            for _layer4 in groupby(_layer3[1], key = lambda x:x[3]):
                level4 = Item(_layer4[0], None)
                level3.Children.Add(level4)
                for _layer5 in groupby(_layer4[1], key = lambda x:x[4]):
                    for _layer6 in groupby(_layer5[1], key = lambda x:x[5]):
                        level5 = Item(_layer5[0], _layer6[0])
                        level4.Children.Add(level5)
    file = open("PythonScripts\\schema.xaml")
    xaml = file.read()
```



```
file.close()
stackPanel = StackPanel()
text = TextBlock()
text.Text = "CSDB Integration Example"
color = SolidColorBrush(Colors.MediumBlue)
text.Foreground = color
text.FontSize = 24
text.HorizontalAlignment = HorizontalAlignment.Center
treview = XamlReader.Parse(xaml)
treview.ItemsSource = items
treview.SelectedItemChanged += Treview_SelectedItemChanged
btnClose = Button(Content = "Close", Width = 150)
btnClose.Click += lambda s, e: ApiWindow.Close()
stackPanel.Children.Add(text)
stackPanel.Children.Add(treview)
stackPanel.Children.Add(btnClose)
scrollView = ScrollViewer()
scrollView.Content = stackPanel
backstage = Fluent.BackstageTabItem()
backstage.Header = "CSDB"
backstage.Content = scrollView
ApiWindow.Backstage.Items.Add(backstage)
#Change selection event
def Treview_SelectedItemChanged(s, e):
    item = e.NewValue
    if item.Uri:
        ApiWindow.OpenDocument(item.Uri)
```



ICN-LPEDIT-A-000000-A-LIBRO-E0600-A-001-01

Fig 5 Creating backstage tab with the file list retrieved from Resource Resolution Service

5.1.1.4 ContentEventArgs
 Provides access to events arguments when interacting with the document content

Namespace: [XmlEditor](#)

Inheritance hierarchy:

[System.Object](#)

[XmlEditor.ContentEventArgs](#)

Syntax:

public sealed class ContentEventArgs



Table 11 Constructors

Name	Description						
ContentEventArgs (XObject, String, String)	<p>Creating new ContentEventArgs object</p> <pre>public ContentEventArgs(XObject xObject, string name, string args)</pre> <p>Parameters</p> <table> <tr> <td>xObject (XObject)</td> <td>Context object</td> </tr> <tr> <td>name (String)</td> <td>Event name</td> </tr> <tr> <td>args (String)</td> <td>Event arguments</td> </tr> </table>	xObject (XObject)	Context object	name (String)	Event name	args (String)	Event arguments
xObject (XObject)	Context object						
name (String)	Event name						
args (String)	Event arguments						

Table 12 Properties

Name	Description
Args	<p>Returns event arguments, defined in the document style markup</p> <pre>public string Args { get;}</pre>
Handled	<p>Gets or sets a value that indicates the present state of the event handling</p> <pre>public bool Handled { get;}</pre>
Name	<p>Returns event name (identifier). Defined in the document style markup</p> <pre>public string Name { get;}</pre>
XObject	<p>Context object</p> <pre>public XObject XObject { get;}</pre>

5.1.1.5 DocumentClosingEventArgs
Provides access to document closing event arguments

Namespace: [XmlEditor](#)

Inheritance hierarchy:

[System.ComponentModel.CancelEventArgs](#)
[XmlEditor.DocumentClosingEventArgs](#)

Syntax:

```
public sealed class DocumentClosingEventArgs : CancelEventArgs
```



Table 13 Constructors

Name	Description
DocumentClosingEventArgs (CloseMode)	Creating new DocumentClosingEventArgs object public DocumentClosingEventArgs(CloseMode closeMode)
	Parameters
	closeMode Document close mode (CloseMode)

Table 14 Properties

Name	Description
CloseMode	Returns document close mode public CloseMode CloseMode { get;}

5.1.1.6 DocumentEventArgs
Provides arguments of DocumentOpened and DocumentCreated events

Namespace: [XmlEditor](#)

Inheritance hierarchy:

[System.EventArgs](#)

XmlEditor.DocumentEventArgs

Syntax:

```
public sealed class DocumentEventArgs : EventArgs
```

Table 15 Constructors

Name	Description
DocumentEventArgs (ApiDocument)	Creating new DocumentEventArgs object public DocumentEventArgs(ApiDocument doc)
	Parameters
	doc (ApiDocument) Document

*Table 16 Properties*

Name	Description
Document	ApiDocument object public ApiDocument Document { get; set;}

5.1.1.7 DocumentFileEventArgs
Provides access to event arguments (closing or saving document)

Namespace: [XmlEditor](#)

Inheritance hierarchy:

[System.Object](#)

[XmlEditor.DocumentFileEventArgs](#)

Syntax:

public sealed class DocumentFileEventArgs

Table 17 Constructors

Name	Description
DocumentFileEventArgs (String)	Creating new DocumentFileEventArgs object public DocumentFileEventArgs(string fileName) Parameters fileName (String) Full path (URI) to the document

Table 18 Properties

Name	Description
FileName	Returns the full path (URI) to the document public string FileName { get;}

5.1.1.8 DocumentSavingEventArgs
Provides arguments for the document saving event

Namespace: [XmlEditor](#)

Inheritance hierarchy:

[System.ComponentModel.CancelEventArgs](#)

[XmlEditor.DocumentSavingEventArgs](#)

Syntax:



public sealed class DocumentSavingEventArgs : CancelEventArgs

Table 19 Constructors



Name	Description
 DocumentSavingEventArgs (String)	Provides saving document info public DocumentSavingEventArgs(string fileName) Parameters fileName (String) Full path (URI) to the document

Table 20 Properties

Name	Description
 FileName	Returns the full path (URI) to the document public string FileName { get;}

5.1.1.9 Selection
Provides access to the document fragment selection model

Namespace: [XmlEditor](#)

Inheritance hierarchy:

[System.Object](#)
XmlEditor.Selection

Syntax:

public class Selection

Table 21 Properties




Name	Description
 Cursor	Returns cursor state public XmlCursor Cursor { get;}
 EXNodes	Returns nodes collection public List<ExtendedXNode> EXNodes { get;}
 IsEmpty	Gets a value that indicates that whether selection is empty public bool IsEmpty { get;}



Table 22 Events

Name	Description
SelectionChanged	Called when the document fragment selection is changed <pre>public event XmlSelectionChangedHandler SelectionChanged { add; remove; }</pre>

Table 23 Methods

Name	Description
Change (ExtendedXNode)	Changes the document fragment selection <pre>public void Change(ExtendedXNode singleNode)</pre> Parameters singleNode Document fragment selection (ExtendedXNode)
Change (ExtendedXNode, ExtendedXNode)	Changes the document fragment selection <pre>public void Change(ExtendedXNode startNode, ExtendedXNode endNode)</pre> Parameters startNode Start selection (ExtendedXNode) endNode End selection (ExtendedXNode)
CheckValidDeletion()	Returns possibility of deleting node set without XSD validation errors <pre>public bool CheckValidDeletion()</pre>
CheckValidInsert (List<XNode>)	Returns possibility of inserting node set without XSD validation errors <pre>public bool CheckValidInsert(List<XNode> nodes)</pre> Parameters nodes (List<>) Node set for inserting



Table 23 Methods (Continued)

Name	Description
☰ CheckValidOuter (String)	Returns possibility of selection encasing with certain element without XSD validation errors public bool CheckValidOuter(string elementName) Parameters elementName Name of the element for encasing (String)
☰ CheckValidReplace (List<XNode>)	Returns possibility of selection replacing with node set without XSD validation errors public bool CheckValidReplace(List<XNode> nodes) Parameters nodes (List<>) Node set for replacing
☰ Delete()	Deletes document fragment selection public void Delete()
☰ Replace (List<XNode>)	Replaces the document fragment selection with the node set public void Replace(List<XNode> nodes) Parameters nodes (List<>) Node set for inserting

5.1.1.9.1 Examples

5.1.1.9.1.1 Encasing fragment selection with verbatimText element

```
btnVerbatim = Fluent.Button(Header = "Verbatim")
btnVerbatim.Size = Fluent.RibbonControlSize.Middle
btnVerbatim.Click += ClickEventVerbatim
```

```
def ClickEventVerbatim(s, e):
```

```
  ApiDocument.TransactionManager.StartTransaction("ApiTransaction")
```

```
  check = ApiDocument.Selection.CheckValidOuter("verbatimText")
```

```
  if check:
```

```
    verbatim = XElement("verbatimText", XAttribute("verbatimStyle", "vs01"))
```

```
    for exnode in ApiDocument.Selection.EXNodes:
```

```
      xnode = exnode.GetNewXNode()
```

```
      verbatim.Add(xnode)
```

```
    listXNodes = List[XNode]()
```

```
    listXNodes.Add(verbatim)
```

```
    ApiDocument.Selection.Replace(listXNodes)
```

```
  ApiDocument.TransactionManager.EndTransaction()
```

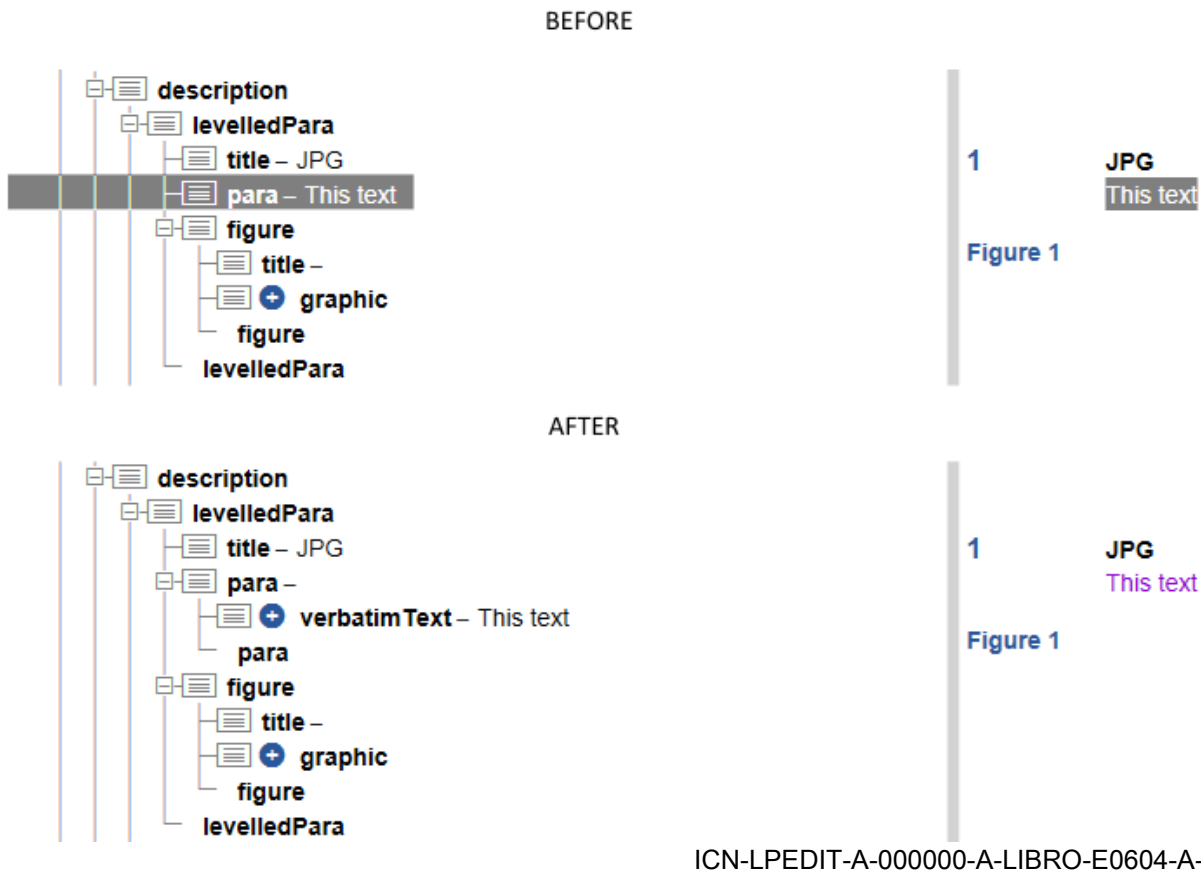


Fig 6 Encasing fragment selection with verbatimText element

5.1.2 Enums

5.1.2.1 CloseMode

Document close mode

- Save (with saving the document)
- NotSave (without saving the document)
- SaveDialog (with opening dialog).

Namespace: [XmlEditor](#)

Syntax:

```
public enum CloseMode : int
```

5.2 XmlEditor.Model (Namespace)

5.2.1 Classes

5.2.1.1 CommandLineOptions

Command line parameters object

Namespace: [XmlEditor.Model](#)



Inheritance hierarchy:



[System.Object](#)

`XmlEditor.Model.CommandLineOptions`

Syntax:

```
public class CommandLineOptions
```

Table 24 Properties

Name	Description
 Document	Full path (URI) to the document <code>public string Document { get; set;}</code>
 readOnly	Gets or sets a document "read only" mode value <code>public bool readOnly { get; set;}</code>

5.2.1.2

ExtendedXNode

The ExtendedXNode class is the extended XObject class. It sets the part of the XText node

Namespace: [XmlEditor.Model](#)

Inheritance hierarchy:

[System.Object](#)

`XmlEditor.Model.ExtendedXNode`

Syntax:

```
public class ExtendedXNode
```

Table 25 Constructors


Name	Description
 ExtendedXNode (XObject)	Used for the following objects: XElement, XAttribute, XComment, XProcessingInstruction <code>public ExtendedXNode(XObject xObject)</code> Parameters xObject (XObject) XML-object



Table 25 Constructors (Continued)

Name	Description				
ExtendedXNode (XObject, XTextPointer)	<p>Creating new ExtendedXNode object</p> <pre>public ExtendedXNode(XObject xObject, XTextPointer textPointer)</pre> <p>Parameters</p> <table> <tr> <td>xObject (XObject)</td> <td>XML-object</td> </tr> <tr> <td>textPointer (XTextPointer)</td> <td>XText part pointer</td> </tr> </table>	xObject (XObject)	XML-object	textPointer (XTextPointer)	XText part pointer
xObject (XObject)	XML-object				
textPointer (XTextPointer)	XText part pointer				

5.2.1.3 XTextPointer
XTextPointer class sets and controls the XText part pointer

Namespace: [XmlEditor.Model](#)

Inheritance hierarchy:

[System.Object](#)
XmlEditor.Model.XTextPointer

Syntax:

```
public class XTextPointer
```

Table 26 Constructors

Name	Description				
XTextPointer (Int32, Int32)	<p>Creating new XTextPointer object</p> <pre>public XTextPointer(int start, int end)</pre> <p>Parameters</p> <table> <tr> <td>start (Int32)</td> <td>Start position</td> </tr> <tr> <td>end (Int32)</td> <td>End position</td> </tr> </table>	start (Int32)	Start position	end (Int32)	End position
start (Int32)	Start position				
end (Int32)	End position				

5.2.1.4 XmlCursor
XmlCursor class provides the cursor object for the XML document. It sets the cursor position in the XML document

Namespace: [XmlEditor.Model](#)


Inheritance hierarchy:

[System.Object](#)
XmlEditor.Model.XmlCursor

**Syntax:**

```
public class XmlCursor
```

Table 27 Constructors

Name	Description								
 XmlCursor (XObject, RPosition, Boolean, Int32)	Creating new XmlCursor object <pre>public XmlCursor(XObject xObject, RPosition relativePosition, bool textAllowed, int offset)</pre> <p>Parameters</p> <table> <tr> <td>xObject (XObject)</td> <td>XML-object</td> </tr> <tr> <td>relativePosition (RPosition)</td> <td>Position relative to XML object</td> </tr> <tr> <td>textAllowed (Boolean)</td> <td>Flag indicating the availability of text insertion</td> </tr> <tr> <td>offset (Int32)</td> <td>Text offset in the text node (0 is for non-text node objects)</td> </tr> </table>	xObject (XObject)	XML-object	relativePosition (RPosition)	Position relative to XML object	textAllowed (Boolean)	Flag indicating the availability of text insertion	offset (Int32)	Text offset in the text node (0 is for non-text node objects)
xObject (XObject)	XML-object								
relativePosition (RPosition)	Position relative to XML object								
textAllowed (Boolean)	Flag indicating the availability of text insertion								
offset (Int32)	Text offset in the text node (0 is for non-text node objects)								

5.2.2 Enums**5.2.2.1 RPosition**

Position relative to attribute or XML tags:

- Before (before start tag)
- First (after start tag)
- Last (before end tag)
- After (after end tag)
- Text (inside the text node)
- Attribute.

Namespace: [XmlEditor.Model](#)

Syntax:

```
public enum RPosition : int
```

6 Multimedia formats support

Elements **graphic**, **symbol** and **multimediaObject** contain attribute **infoEntityIdent** with reference to unparsed entity. Each entity should have a reference to the notation indicating entity type (multimedia type).

The path to the public notations list:

- DISK:\Program files (x86)\Libroplanta\S1000D XML Editor\DocTypes\notations.xml



The public notations list can refer to the display template that must be stored in the folder with the same name as the template name. The base path to templates:

- DISK:\Program files (x86)\Libroplanta\S1000D XML Editor\Multimedia\[Template]

Users can add their own display templates and corresponding folders for their own multimedia types or reuse existing templates.

Note

The application uses Chromium web-browser for displaying graphics and multimedia. Supported formats: <https://www.chromium.org/audio-video>

When adding user display template, it may be necessary to modify the file:

- DISK:\Program files (x86)\Libroplanta\S1000D XML Editor\Multimedia\Service\multimediaWrapper.js

To evaluate CGM supporting you can insert reference to VizEx View HTML5 ("http://www.cgmlarson.com/demos/jsViewer/VizExView.jsz") into file: DISK:\Program files (x86)\Libroplanta\S1000D XML Editor\Multimedia\CGM\index.htm

Preliminarily read the license agreement on VizEx View HTML5 on the owner's website.

7 URL-schema support

Libroplanta S1000D XML Editor provides URL-schema registration for running editor when user clicks on a URL-link containing this URL-schema, from any application.

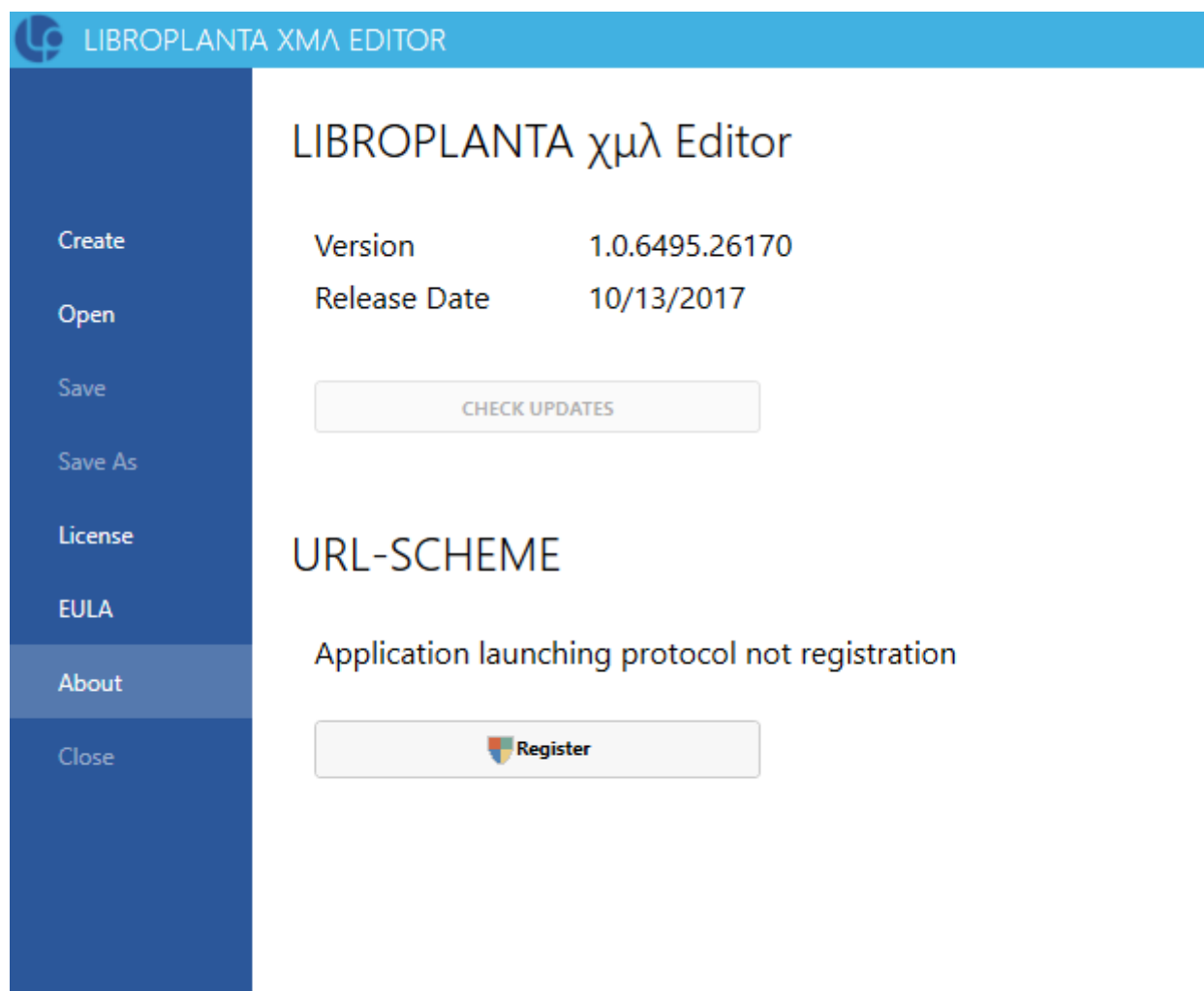
URL-schema name: **lp-xmleditor**.

HTML code fragment for running Libroplanta S1000D XML Editor and passing the file name as the argument to it:

```
<html>  
  <a href="lp-xmleditor:?http://myhost//DMC-BRAKE-AAA-  
D00-00-00-00AA-00WA-D_001-00_ru-RU.xml">DMC-BRAKE-AAA-  
D00-00-00-00AA-00WA-D</a>  
</html>
```

The hyper link parameter contains the Libroplanta S1000D XML Editor URL-schema and query arguments for passing the path to file to be opened.

The protocol.exe utility provides URL-schema registration. To run this utility, go to the "About" tab in the main window of Libroplanta S1000D XML Editor and click "Register" (Fig 7).



ICN-LPEDIT-A-000000-A-LIBRO-E0610-A-001-01

Fig 7 URL-schema registration

8 Elements identifiers naming and assigning rules

User can change the elements identifiers naming and assigning rules for each ASD S1000D Issue.

Path to the settings file for elements identifiers naming and assigning rules:

- DISK:\Program files (x86)\Libroplanta\S1000D XML Editor\DocTypes\[S1000D Issue]\ldRefs\ldRefs.xml

9 User smart tags

User can change and add own smart tags for each ASD S1000D Issue.



Path to the settings file for smart tags:

- DISK:\Program files (x86)\Libroplanta\S1000D XML Editor\DocTypes\[S1000D Issue]\SmartTags\Settings.xml

10 User context rules

User context rules are the rules for creating the context menu in accordance with the document fragment selection context.

Path to the settings file for user context rules:

- DISK:\Program files (x86)\Libroplanta\S1000D XML Editor\DocTypes\[S1000D Issue]\Formatting\Rules.xml

11 Document display styles

User can change the document display styles for each ASD S1000D Issue and for each document type.

Base path to the display style files:

- DISK:\Program files (x86)\Libroplanta\S1000D XML Editor\DocTypes\[S1000D Issue]\Styles