

Using RealObjects edit-on® Pro 2.x with IBM Lotus Domino 5.x Server

Rev. 1.1 – 2002-12-18

Copyright © 2000-2002 RealObjects GmbH, Saarbrücken, Germany.
All Rights Reserved.

edit-on® is a registered trademark of RealObjects GmbH, Saarbrücken, Germany.

REALOBJECTS

Gesellschaft für objektorientierte Softwareentwicklung und IT Beratung mbH

Tel.: +49 (0) 681 985 790
Fax: +49 (0) 681 985 7929
E-mail: info@realobjects.de
Internet: www.realobjects.de

1. General

To demonstrate how to use RealObjects *edit-on® Pro 2.x* with IBM Lotus Domino 5.x Server we provide a self-contained demonstration database `RTE.nsf` with all the elements needed to run *edit-on® Pro 2.x* in a simple application. The application allows the user to create, edit and review documents which can also be saved to the demonstration database.

In order to run the application you need an *edit-on® Pro 2.x* installation package (Version 2.1.119.FIX.2), including a valid 30-day trial or full license key. Please feel free to request an *edit-on® Pro 2.x* trial version at www.realobjects.de/trial

This white paper assumes that you have some working knowledge of Lotus Domino 5.x Server, Lotus Domino Designer and JavaScript.

2. System requirements

- IBM Lotus Domino 5.x Server or newer
- One of the following browsers
 - Microsoft Internet Explorer 4.x
 - Microsoft Internet Explorer 5.x
 - Microsoft Internet Explorer 6.x
 - Netscape Navigator 4.7 or newer
 - Netscape 6.1 or newer
 - Mozilla 1 or newer
 - Other browsers with Java 1.1 applet support

3. Get the application running

There are only five steps to get the application running:

1. Unzip the *edit-on® Pro 2.x* installation package. For further information please review the `readme.txt` and Integration Manual, both are part of the installation package.
2. Copy the complete `eopro` folder of the *edit-on® Pro 2.x* installation package (including all sub-folders and files) into the `html` folder within the Lotus Domino Server, e.g. `<hostname>:\Lotus\Domino\Data\domino\html\eopro`. Please note, `eopro` is the applet's **CODEBASE** which is initialised through the *edit-on® Pro 2.x* `<APPLET>` tag, see also chapter 4/2a below.
3. Copy the *edit-on® Pro 2.x* trial or full license key file (`licensekey.xml`) into the `eopro` folder within the Lotus Domino Server, e.g. `<hostname>:\Lotus\Domino\Data\domino\html\eopro\licensekey.xml`
4. Copy the demonstration database (`RTE.nsf`) into the `Data` folder within the Lotus Domino Server, e.g. `<hostname>\Lotus\Domino\Data\RTE.nsf`.
5. To run the application from the server, launch a web browser and enter <http://<hostname>/RTE.nsf>. To run it locally on the server, enter <http://127.0.0.1/RTE.nsf> or <http://localhost/RTE.nsf>.

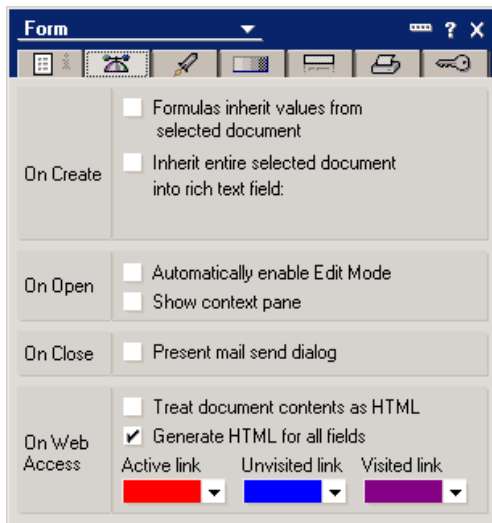
4. Developer information

Within this chapter you will find detailed information on how *edit-on® Pro 2.x* can be integrated into a Lotus Notes web application, here the self-contained demonstration database `RTE.nsf`. Following steps have to be done:

1. Create a form

Open Domino Designer and create a new form. Please note, for each form field which should be editable by using the editor applet, you have to initialise a separate instance of *edit-on® Pro 2.x*.

IMPORTANT: Within the form properties-box, select “Generate HTML for all fields”

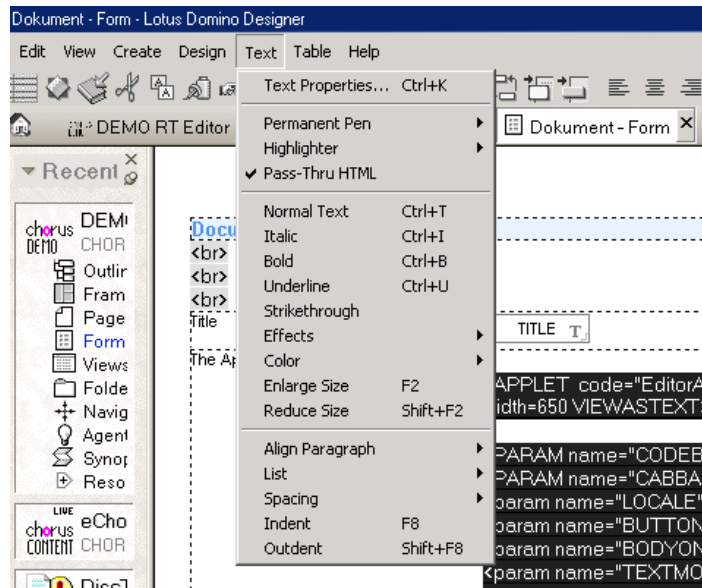


2. Insert *edit-on® Pro 2.x* for use as a rich text editor

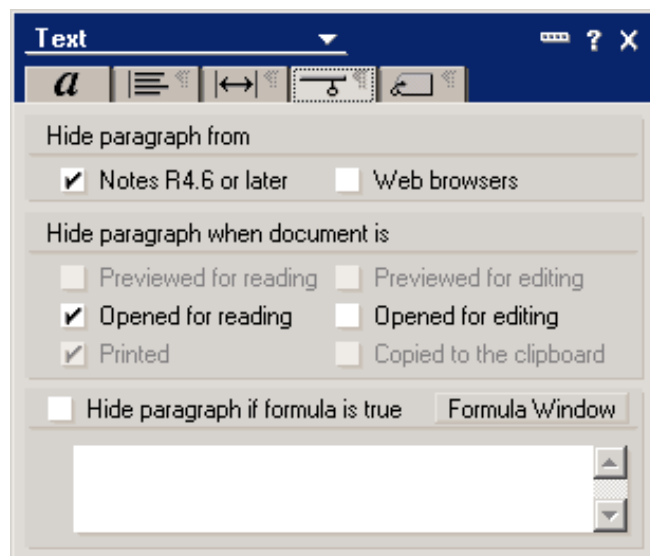
a) Use *edit-on® Pro* integration manual in order to specify the *edit-on® Pro* functionality according to your needs. Following you will find an example of an `<APPLET>` tag as used in the self-contained demonstration database `RTE.nsf`:

```
<APPLET code="EditorApplet" archive="edit-on-pro.jar,ssce.jar" height=350 name=EditorTxt1
width=650 VIEWASTEXT MAYSCRIPT>
<PARAM name="CODEBASE" value="/eopro/">
<PARAM name="CABBASE" value="edit-on-pro.cab,ssce.cab">
<PARAM name="LOCALE" value="de_DE">
<PARAM name="TOOLBARURL" value="toolbar-sample.xml">
<PARAM name="BODYONLY" value="true">
<PARAM name="TEXTMODE" value="FALSE">
<PARAM name="SOURCEVIEW" value="TRUE">
<PARAM name="SIMPLETABLE" value="TRUE">
<PARAM name="TABLENBSPFILL" value="true">
<PARAM name="DEFAULTFONTFACE" value="Arial">
<PARAM name="DEFAULTFONTSIZE" value="10">
<PARAM name="DEFAULTFONTCOLOR" value="#000000">
<PARAM name="DEFAULTLINKCOLOR" value="#000000">
<PARAM name="DEFAULTBACKGROUND" value="#FFFFFF">
<PARAM name="ONEDITORLOADED" value="setApplet">
</APPLET>
```

b) After finishing the HTML code, select the code and mask it as Pass-Thru-HTML (choose from Lotus Designer Menu `>Text>Pass-Thru HTML`).



c) Hide the code in the text properties-box (press Ctrl + K) from the Notes-Client and when the document is opened for reading as shown in the picture below.



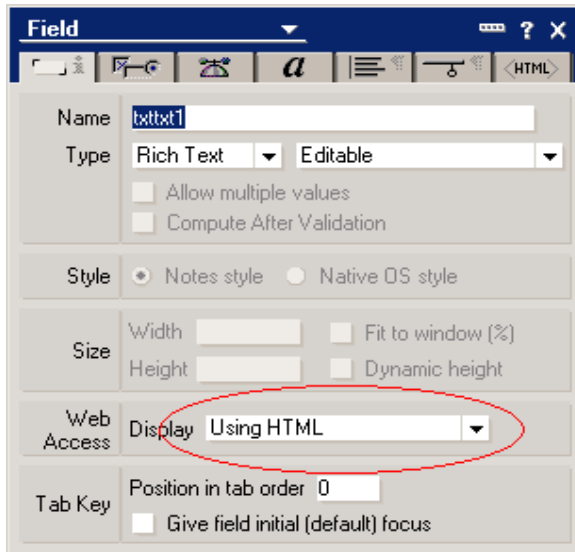
These settings prevent errors when the document is opened in read-mode or with the IBM Lotus Notes client.

3. Further fields in the form

a) Create the rich text field in which you want to save the content created with the edit-on® Pro 2.x applet, e.g. `txttxt1`



IMPORTANT: You have to select “Using HTML” as WebAccess-display-options in the field properties-box.



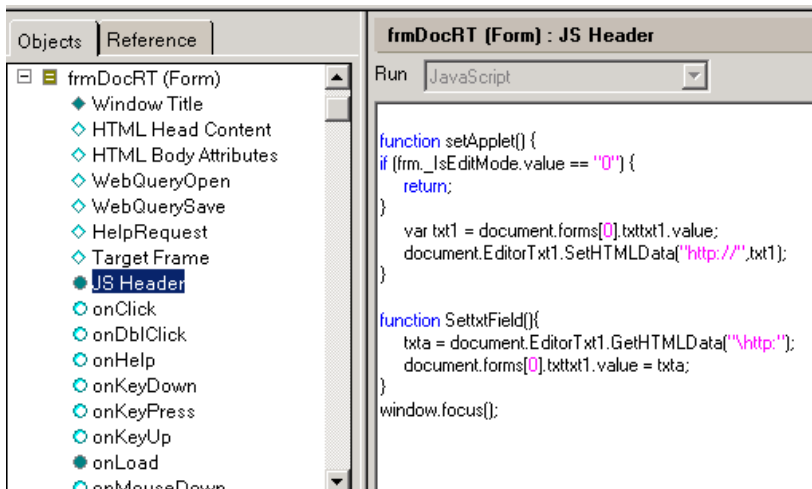
b) For processing the form, you have to create a hidden field, computed for display, e.g.:

Name: `_IsEditMode`
 DefaultValue: `@If(@IsDocBeingEdited; "1"; "0")`

4. JavaScript-Code

a) The setApplet-function

Insert in the “JS Header” of the form the following JavaScript-Code:



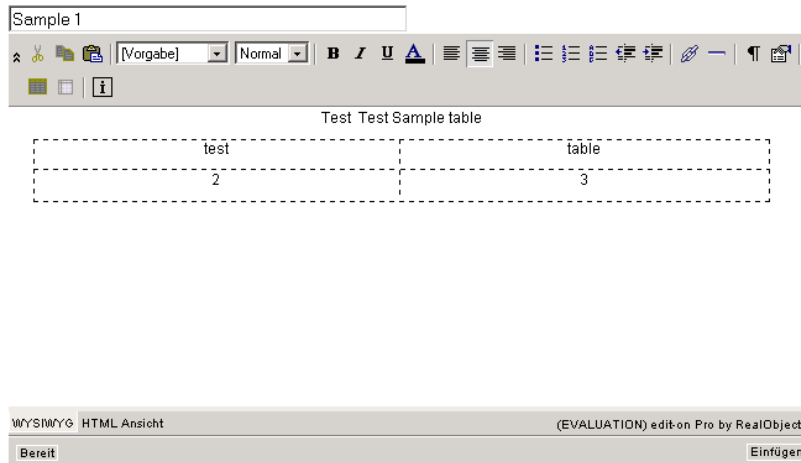
The code:

```
function setApplet() {
if (document.forms[0]._IsEditMode.value == "0") {           //name of the hidden field – see above
    return;
}
    var txt1 = document.forms[0].txttxt1.value;             //name of the rich text field – see above
    document.EditorTxt1.SetHTMLData("http://",txt1);       // name of the applet tag – see above
}
```

IMPORTANT: Make sure that following line is available in the <APPLET> tag:

```
<PARAM name="ONEDITORLOADED" value="setApplet">
```

If you create a new or edit an existing document, this PARAM-value calls the setApplet function shown above. This function inserts the content of the rich text field into the editor. When you now create or edit a document using a browser, the editor will be displayed and the content of the field `txttxt1` will be displayed within the applet.



b) The SettxtField-function

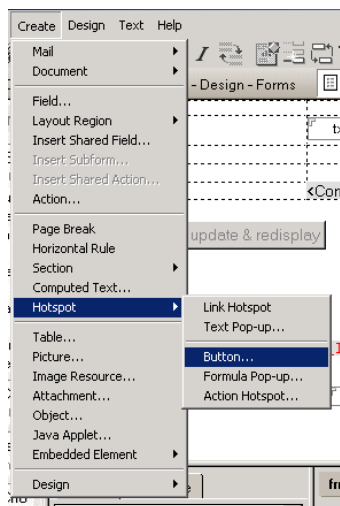
To write the edited content back into the rich text field, insert the following JavaScript-Code into the JS Header.

```
function SettxtField(){
    var txt = document.EditorTxt1.GetHTMLData("\http:");
    document.forms[0].txttxt1.value = txt;
}
```

This function is called when you save the document.

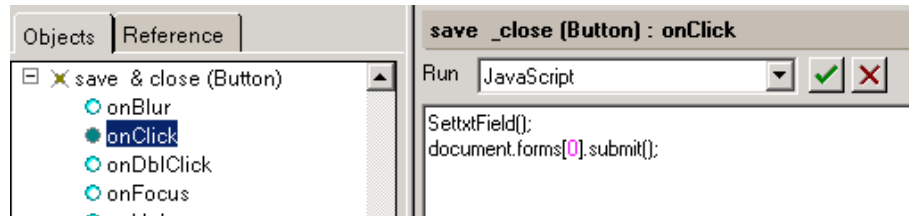
c) Save the document

Create a "Hotspot Button" (Lotus Designer Menu >Create>Hotspot>Button...).

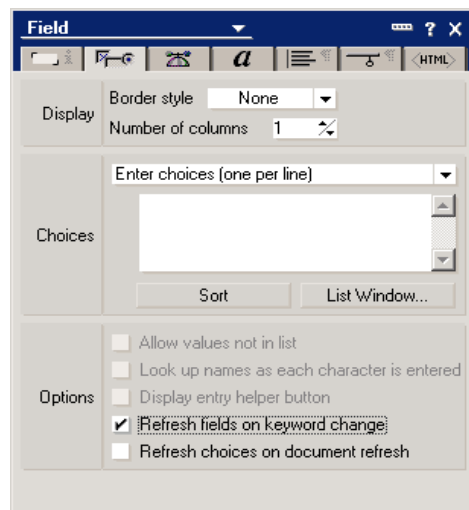
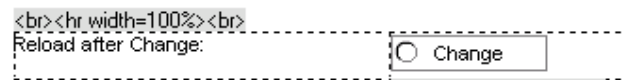


Choose “Run JavaScript” in the programmers-pane, then insert the following code in the “onClick”-Event of this Button:

```
SettxtField(); //writes the RT-applet-content to the RT-field (see JS-Header)
document.forms[0].submit(); //saves the document back to disk
```



d) Using “Refresh fields on keyword change”



If you use this option, you have to set the “SettxtField();” Java-Script into “onChange” Section.

