# NativeDoc 3 User Manual

# Introduction

# Welcome to NativeDoc 3

**Welcome to NativeDoc 3!**

NativeDoc is a source code documentation tools for LiveCode.

It has been developed in focus of respecting already existing standard already available in other programming languages (DoxyGen and JavaDoc) in order to be easy to learn/use.

Almost all DoxyGen and JavaDoc tutorials about the writing of documentation block are applicable to NativeDoc, this documentation covers the configuration and the generation of the documentation.

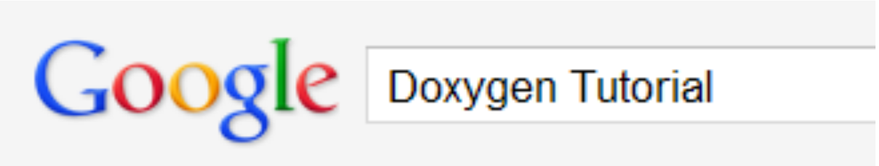
This guide will help you to get the most from NativeDoc 3.0 easily, it covers almost all the features of NativeDoc 3.

# Write documentation for your code

## JavaDoc/DoxyGen

How to write NativeDoc documentation blocks.

### Learn Doxygen or JavaDoc

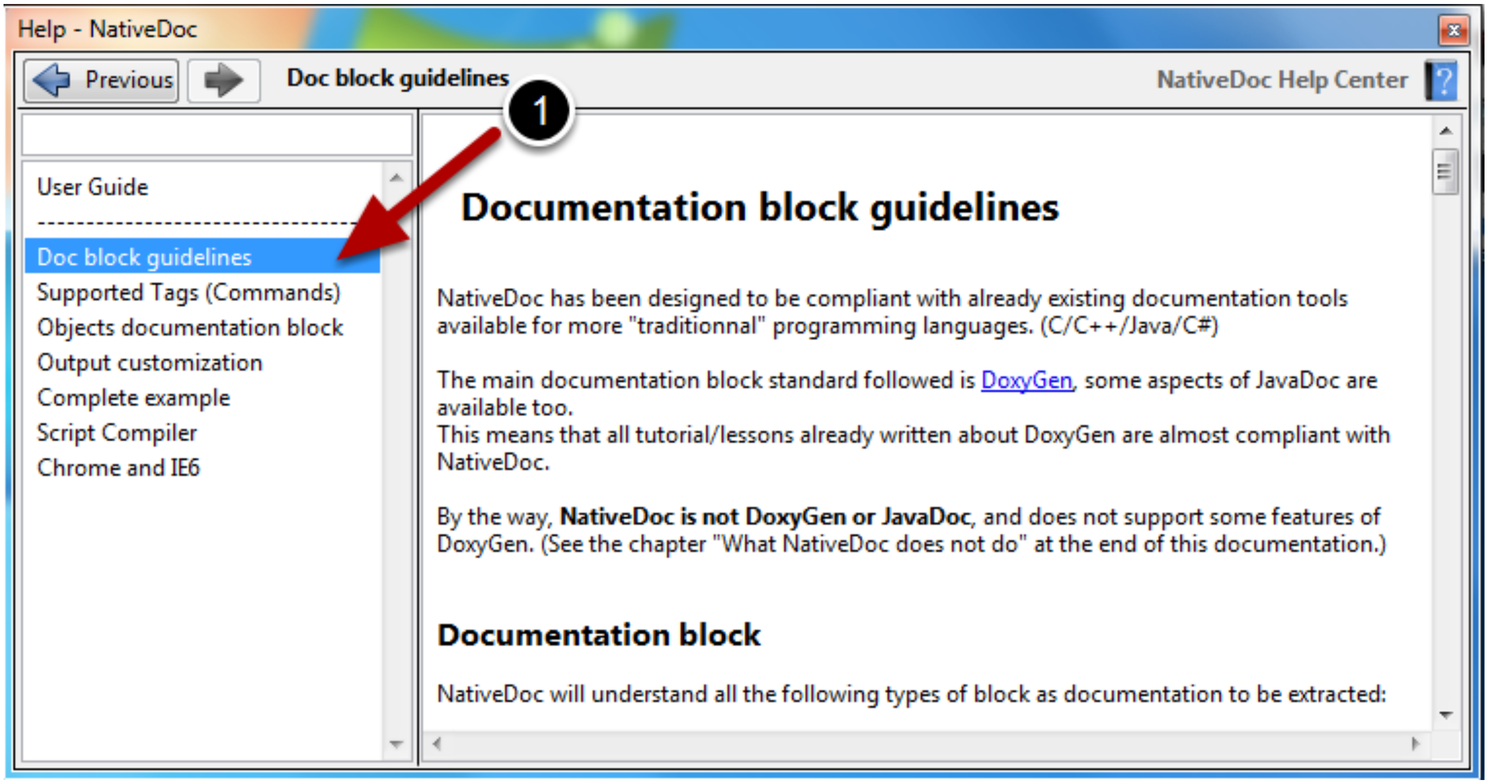


Doxygen Tutorial

Doxygen/JavaDoc are a source code documentation language. They are widely used for C/C++/Java and a lot of other languages by a lot of developper in the world.

NativeDoc is compliant with a lot of Doxygen and JavaDoc instruction, so before starting using NativeDoc please find on google a Doxygen or JavaDoc tutorial in your language!

### The NativeDoc Help Center



**Documentation block guidelines**

NativeDoc has been designed to be compliant with already existing documentation tools available for more "traditionnal" programming languages. (C/C++/Java/C#)

The main documentation block standard followed is DoxyGen, some aspects of JavaDoc are available too.
This means that all tutorial/lessons already written about DoxyGen are almost compliant with NativeDoc.

By the way, **NativeDoc is not DoxyGen or JavaDoc**, and does not support some features of DoxyGen. (See the chapter "What NativeDoc does not do" at the end of this documentation.)

**Documentation block**

NativeDoc will understand all the following types of block as documentation to be extracted:

The list of supported block style and what NativeDoc handles is located inside the NativeDoc Help center. (1)

The help center is intended to help you look quickly what NativeDoc supports and how to write documentation block for NativeDoc.

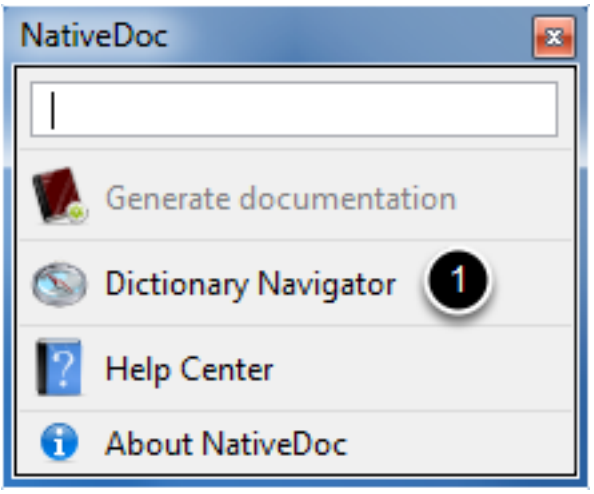
# NativeDoc 3 Documentation Navigator

## Introduction

The NativeDoc documentation navigator is a tool allowing you to browse/search your source code documentation.
It can also import external documentation, allowing you to have quickly access to external LiveCode library API instantly.

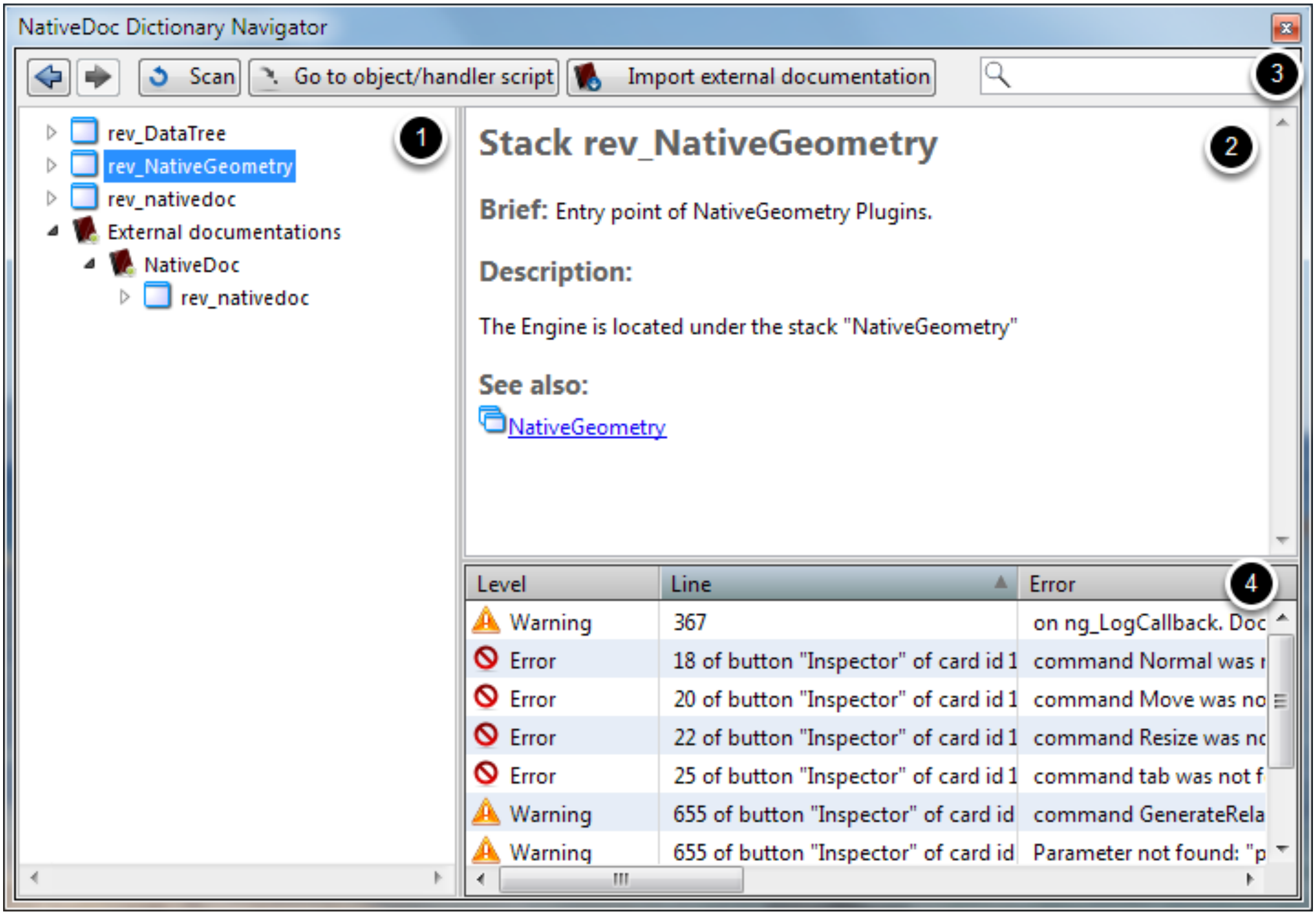
This documentation is using NativeSoft **NativeGeometry** as sample application. NativeGeometry is a geometry manager for LiveCode.

## Opening the Navigator



Simply click "Dictionary Navigator"

## The Navigator window



**Stack rev_NativeGeometry**

**Brief:** Entry point of NativeGeometry Plugins.

**Description:**

The Engine is located under the stack "NativeGeometry"

**See also:**

NativeGeometry

| Level | Line | Error |
|---|---|---|
| Warning | 367 | on ng_LogCallback. Doc |
| Error | 18 of button "Inspector" of card id 1 | command Normal was r |
| Error | 20 of button "Inspector" of card id 1 | command Move was no |
| Error | 22 of button "Inspector" of card id 1 | command Resize was nc |
| Error | 25 of button "Inspector" of card id 1 | command tab was not f |
| Warning | 655 of button "Inspector" of card id | command GenerateRela |
| Warning | 655 of button "Inspector" of card id | Parameter not found: "p |

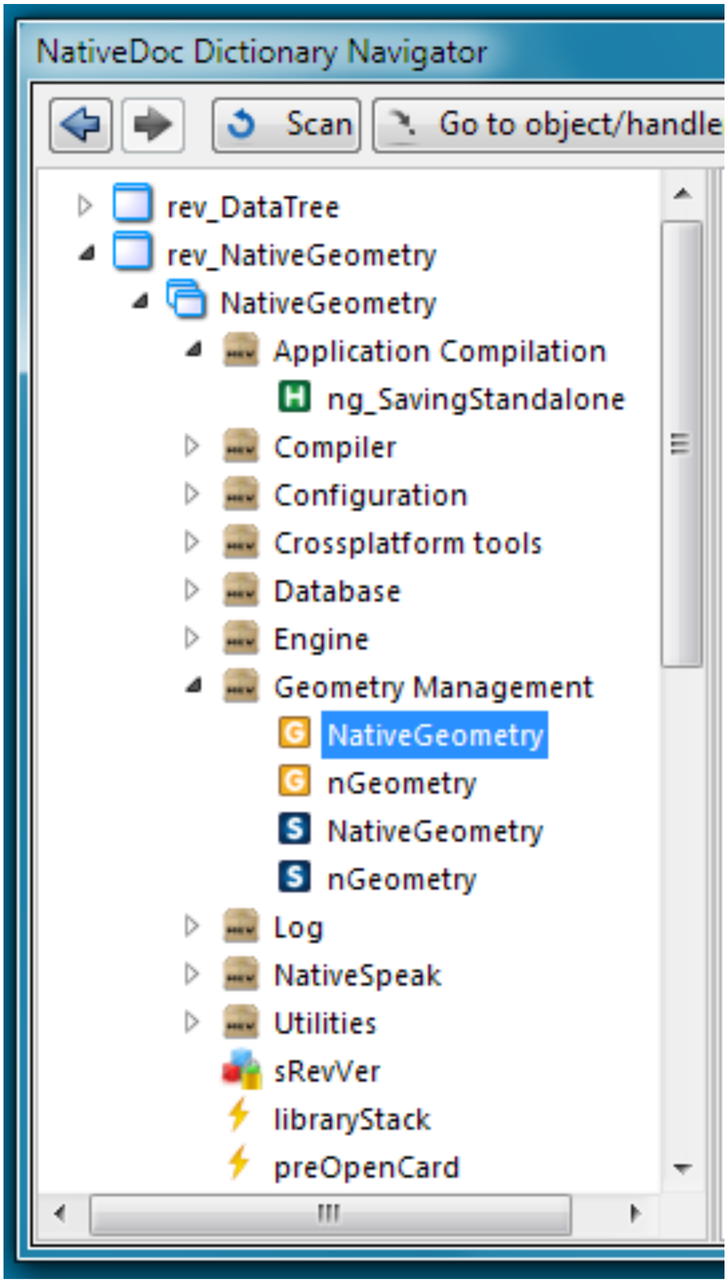
This screenshot show the main window of the NativeDoc Dictionary Navigator.

Explanation of areas:
- 1: Tree opened stacks, all documented handlers will appear here.
- 2: The documentation of the selected element.
- 3: The search bar.
- 4: List of NativeDoc parser errors for the selected object and its childs.

**Note:** NativeDoc display by default only documented element inside the Tree, it is possible to configure NativeDoc to display all elements in the menu "Scan" (click "Scan" button to open the menu).

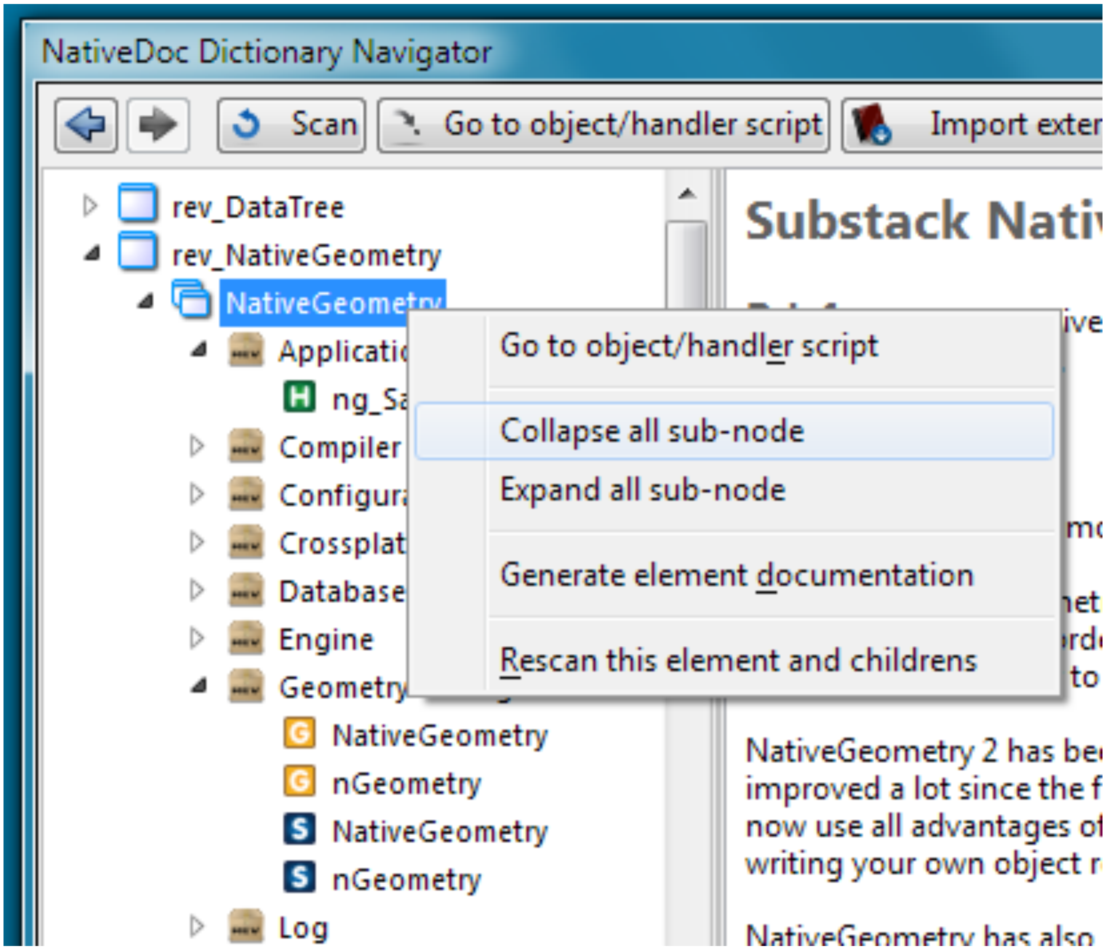
**Note about the Tree:** Stacks displayed inside the tree are the ones displayed in the LiveCode application browser. So it depends on the value of "LiveCode UI Elements in Lists" setting.

## Navigate with the tree



It is possible to browse the documentation using the navigation tree.

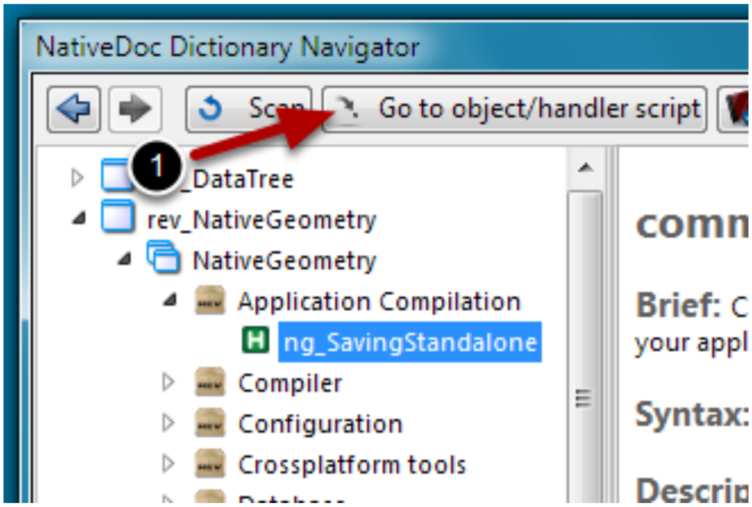
The tree can be navigated with the keyboard using the "Left/Right/Up/Down" arrows, double clicking an element will expand or collapse it.



The tree also have a contextual menu. To access to this menu, right click a node on the tree.

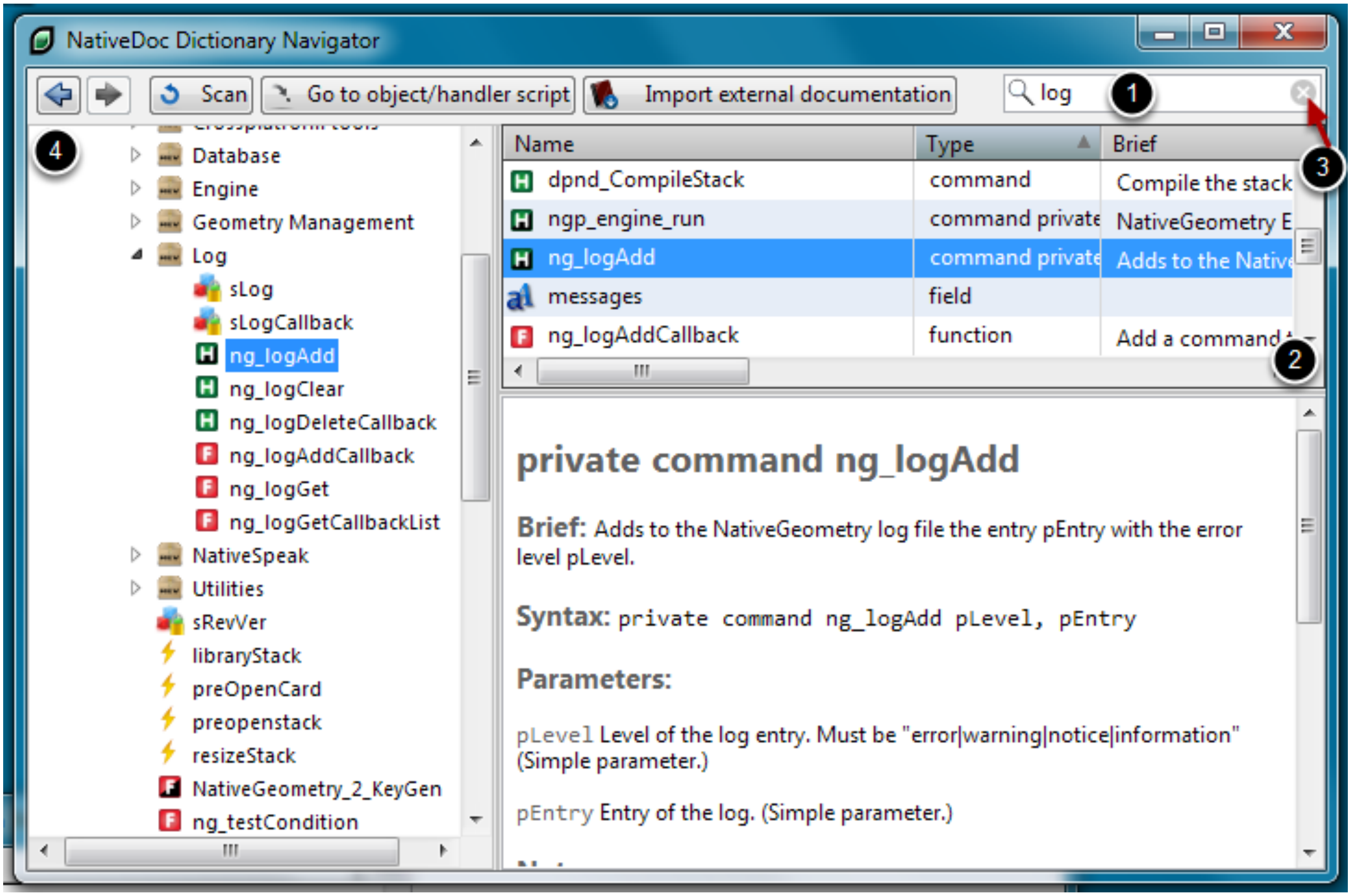
This menu can change depending which node you clicked.

## Select an object/Edit script



The "Go to object/handler script" have the following behavior:
- For an object: The object is opened and selected.
- For an handler: The script editor is opened at the line where the handler is.

## Search



| Name | Type | Brief |
|---|---|---|
| dpnd_CompileStack | command | Compile the stack |
| ngp_engine_run | command private | NativeGeometry E |
| ng_logAdd | command private | Adds to the Native |
| messages | field | |
| ng_logAddCallback | function | Add a command |

### private command ng_logAdd

**Brief:** Adds to the NativeGeometry log file the entry pEntry with the error level pLevel.

**Syntax:** `private command ng_logAdd pLevel, pEntry`

**Parameters:**

`pLevel` Level of the log entry. Must be "error|warning|notice|information" (Simple parameter.)

`pEntry` Entry of the log. (Simple parameter.)

NativeDoc 3.0 allows you to search through all the documentation opened in the Navigator. (Including externals)

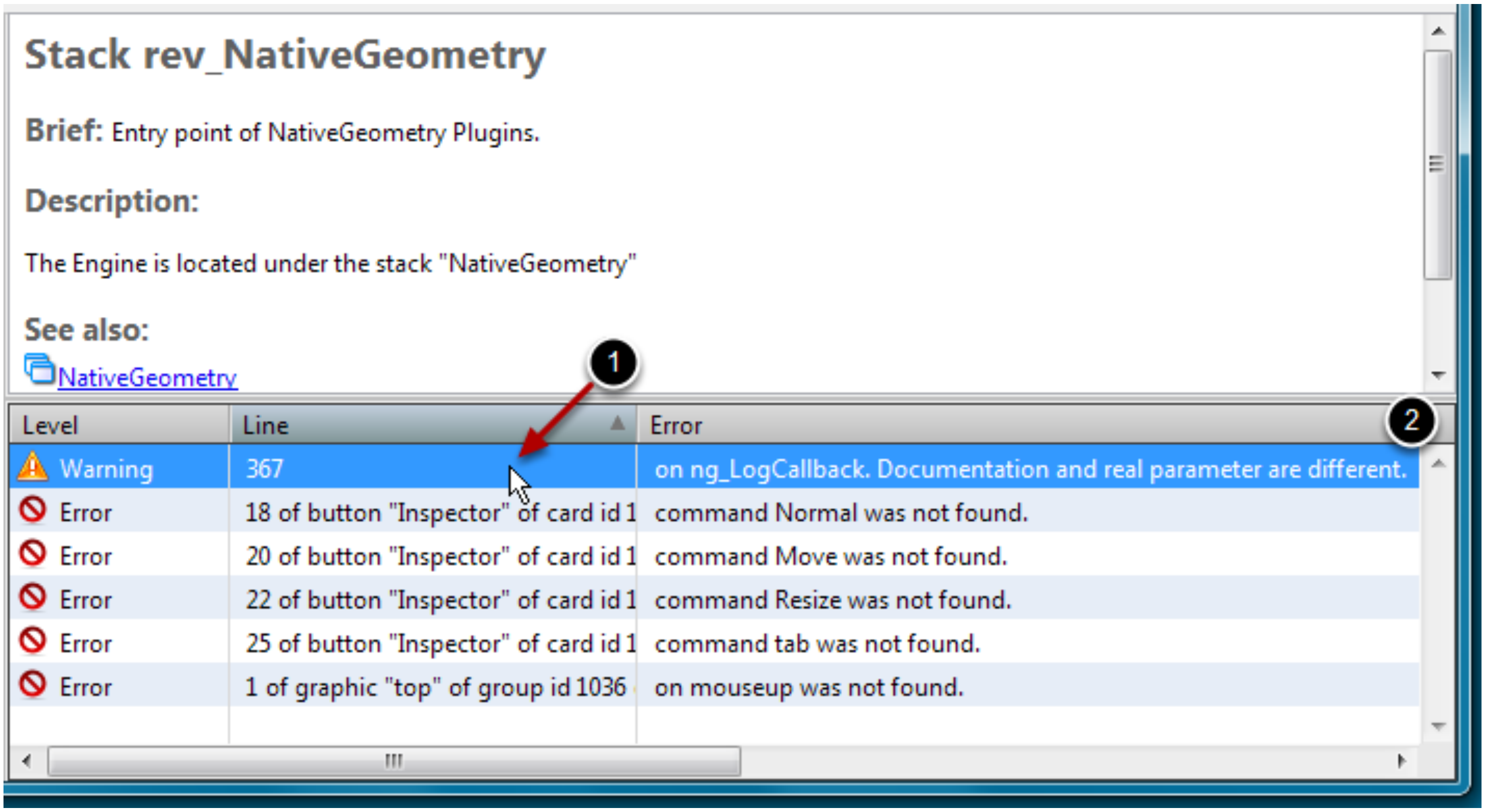
The search engine will look in **the documentation title** and **inside the documentation content**.

Enter the text that you wish search in the search bar (1), then the result window will appear and display instantly results (2).

**Tip:** To select an element in the Navigation tree (4), double click on the element in the search results.

To close the search, click the "Close" icon in the search bar. (3)

## View documentation parsing errors



Stack rev_NativeGeometry

Brief: Entry point of NativeGeometry Plugins.

Description:

The Engine is located under the stack "NativeGeometry"

See also:

NativeGeometry

| Level | Line | Error |
|---|---|---|
| ⚠ Warning | 367 | on ng_LogCallback. Documentation and real parameter are different. |
| 🚫 Error | 18 of button "Inspector" of card id 1 | command Normal was not found. |
| 🚫 Error | 20 of button "Inspector" of card id 1 | command Move was not found. |
| 🚫 Error | 22 of button "Inspector" of card id 1 | command Resize was not found. |
| 🚫 Error | 25 of button "Inspector" of card id 1 | command tab was not found. |
| 🚫 Error | 1 of graphic "top" of group id 1036 | on mouseup was not found. |

NativeDoc 3.0 displays the list of error that the NativeDoc parser encountered.

This enables you to quickly fix problems in your documentation.

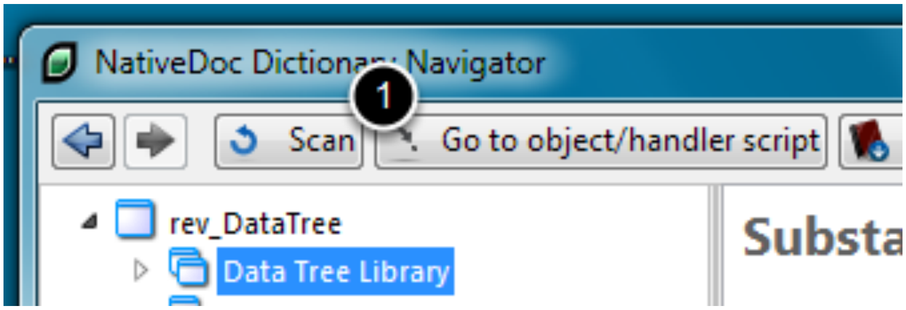
Double click on an element in the documentation parsing error window (1) to open the script editor at the documentation block. So you can fix the error.

## Refreshing the documentation

Since NativeDoc 3, you do not need to refresh manually the documentation list; NativeDoc analyze automatically updated scripts and refresh the Navigator.

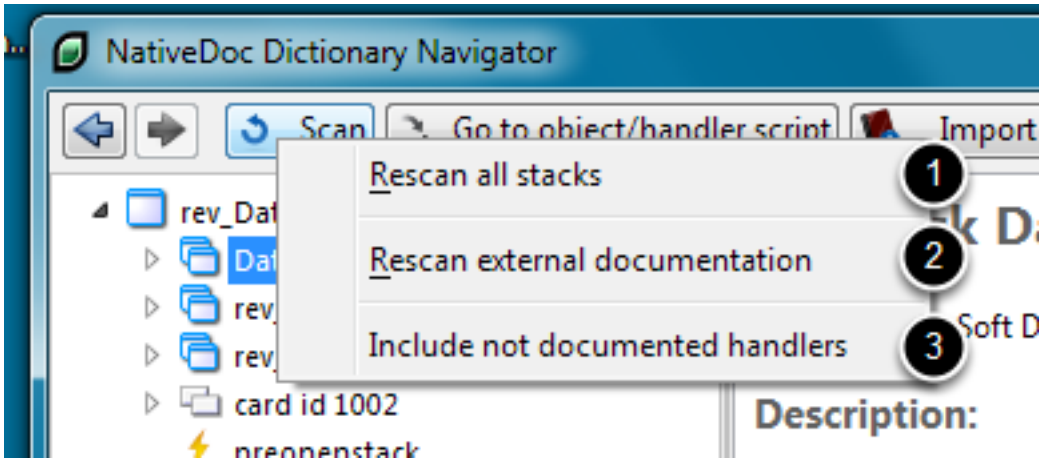
But the automatic scan is not fault-free, so you may have to manually refresh the Navigator content.

### The Scan button



Click button "Scan" in order to popup the "Scan" menu.

### The Scan menu


NativeDoc Dictionary Navigator
Scan
Go to object/handler script
Import
Rescan all stacks
1
Rescan external documentation
2
Include not documented handlers
3
rev_Dat
Dat
rev
rev
card id 1002
preopenstack
Description:

**Rescan all stacks (1):**

Force NativeDoc to re-analyze all your opened LiveCode stacks.

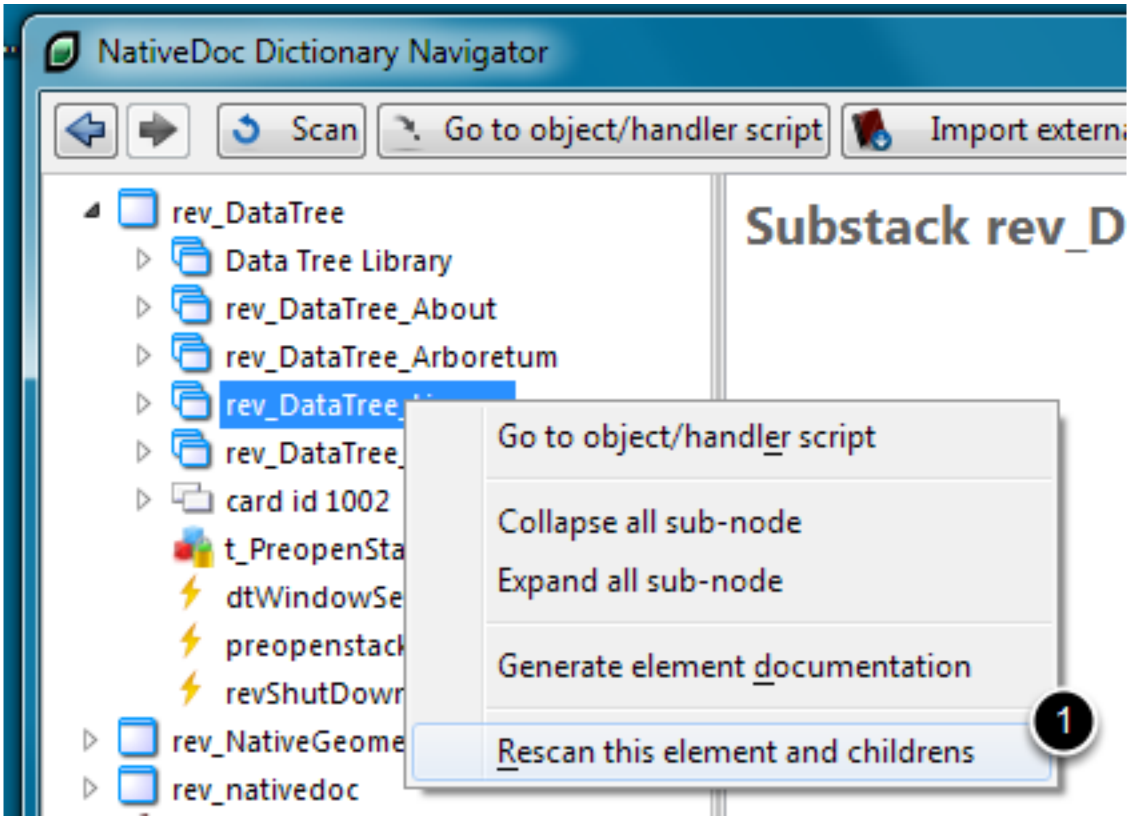
**Rescan external documentation (2) :**

Rescan all external documentation. Useful if you changed the external documentation.

**Include not documented handlers (4) :**

Click to enable/disable the listing of all handlers inside the navigator.

## Rescan a stack using the contextual menu



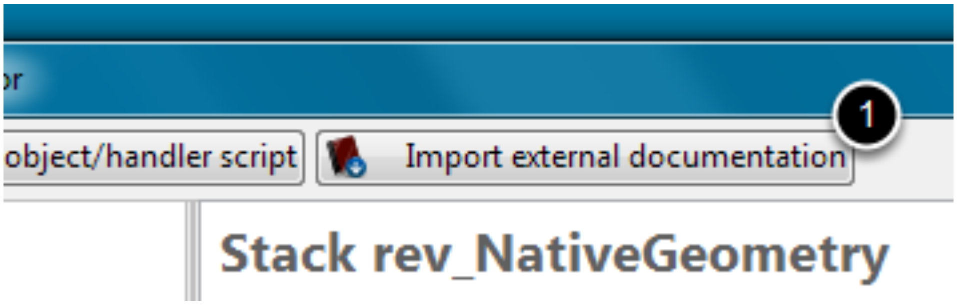
You can also rescan one element and all its childrens using the contextual menu on the tree.

Click (1) to refresh the documentation.

## Add an external documentation

Add an external documentation inside the Navigator. Useful to import external library documentation.

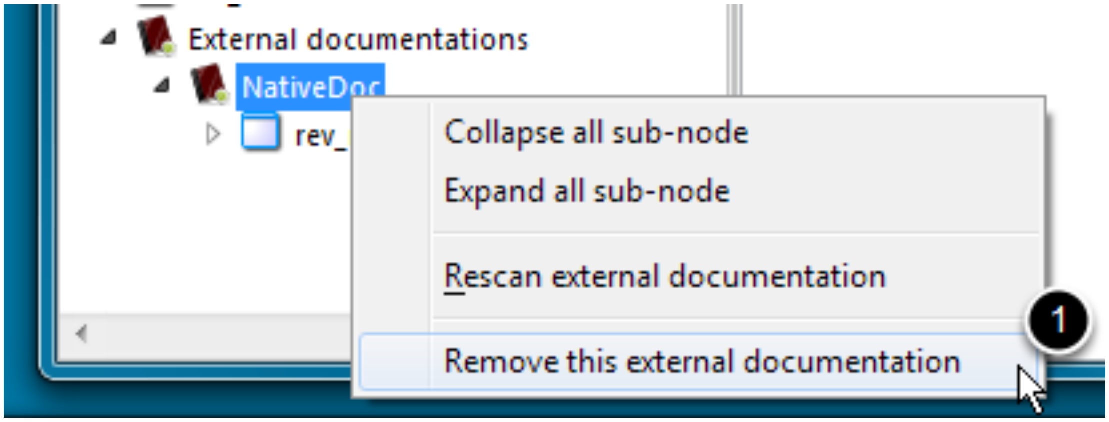
### Import an external documentation



Click "Import external documentation" then select the NativeDoc documentation XML file.

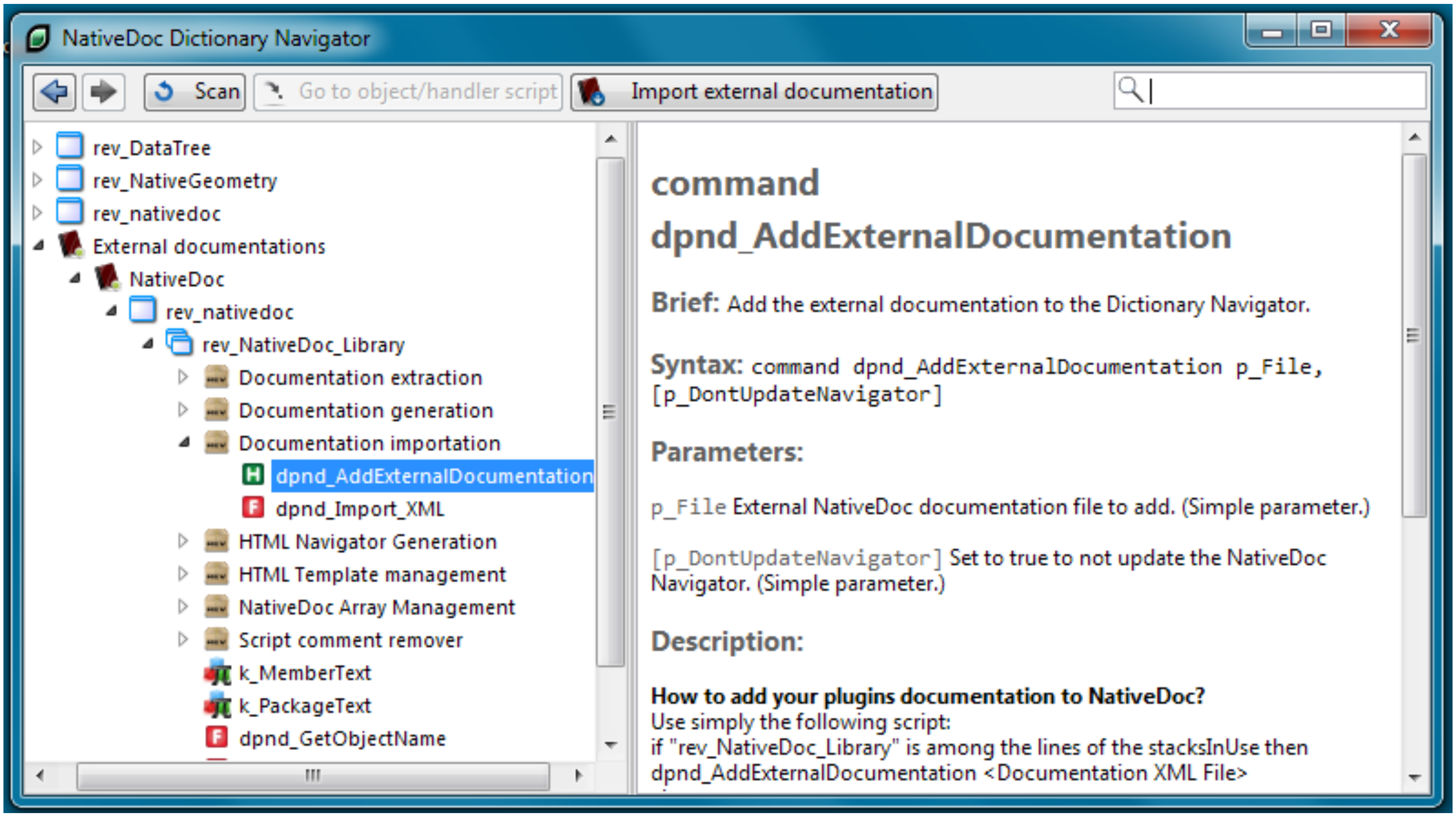
NativeDoc will import it and the documentation will be displayed under the "External documentations" section in the tree.

### Remove an external documentation



To remove an external documentation, right click on the external documentation in the tree then click "Remove this external documentation" (1).

## Import automatically an external documentation



command
dpnd_AddExternalDocumentation

**Brief:** Add the external documentation to the Dictionary Navigator.

**Syntax:** `command dpnd_AddExternalDocumentation p_File, [p_DontUpdateNavigator]`

**Parameters:**

`p_File` External NativeDoc documentation file to add. (Simple parameter.)

`[p_DontUpdateNavigator]` Set to true to not update the NativeDoc Navigator. (Simple parameter.)

**Description:**

**How to add your plugins documentation to NativeDoc?**
Use simply the following script:

```
if "rev_NativeDoc_Library" is among the lines of the stacksInUse then
dpnd_AddExternalDocumentation <Documentation XML File>
```

NativeDoc can import automatically an external documentation at its startup. To do so read the help of the NativeDoc command "dpnd_addExternalDocumentation".

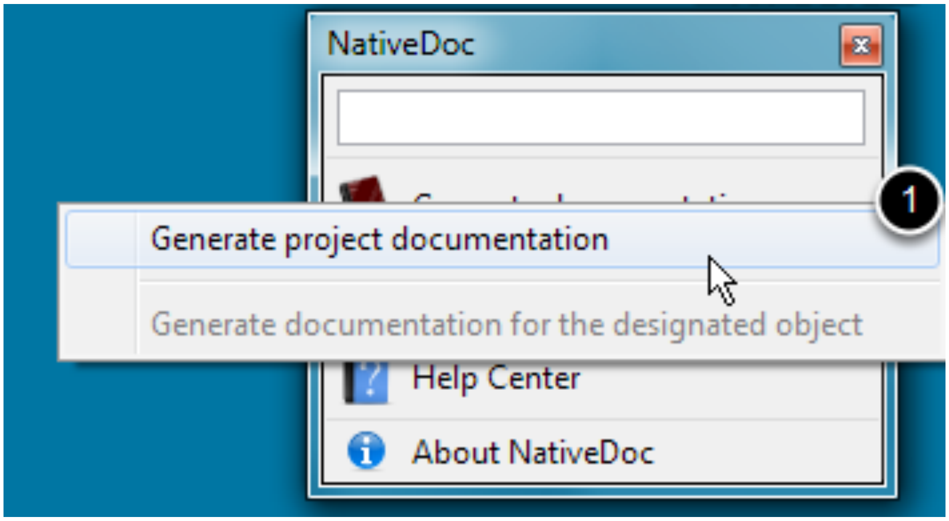
# Documentation Exportation

## First mode: Project mode

NativeDoc 3.0 has a new documentation generation window, that is more straighforward and more complete.

**The project mode:** Allows you to select multiple mainstack, objects and generate an entire documentation.

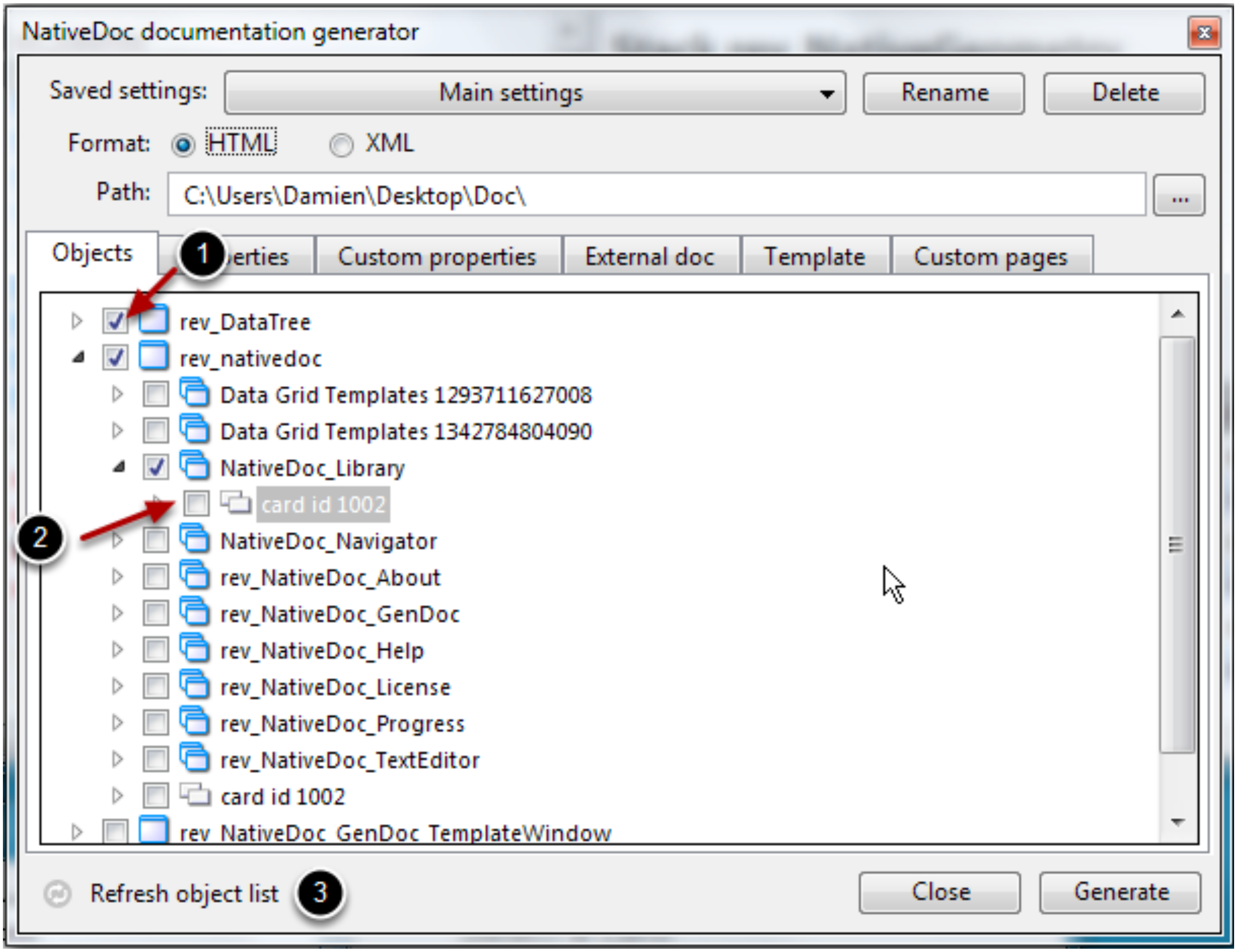
### Generate a documentation project


NativeDoc
1
Generate project documentation
Generate documentation for the designated object
Help Center
About NativeDoc

To open the project documentation generation window, click "Generate documentation" in the NativeDoc palette.

The documentation generator window will open in **"Project mode".**

## Select objects that you want to include in your documentation project



You can select objects that you want to include in the documentation generation.

Simply check objects that you want in your documentation (1) and uncheck ones that you want to exclude (2).

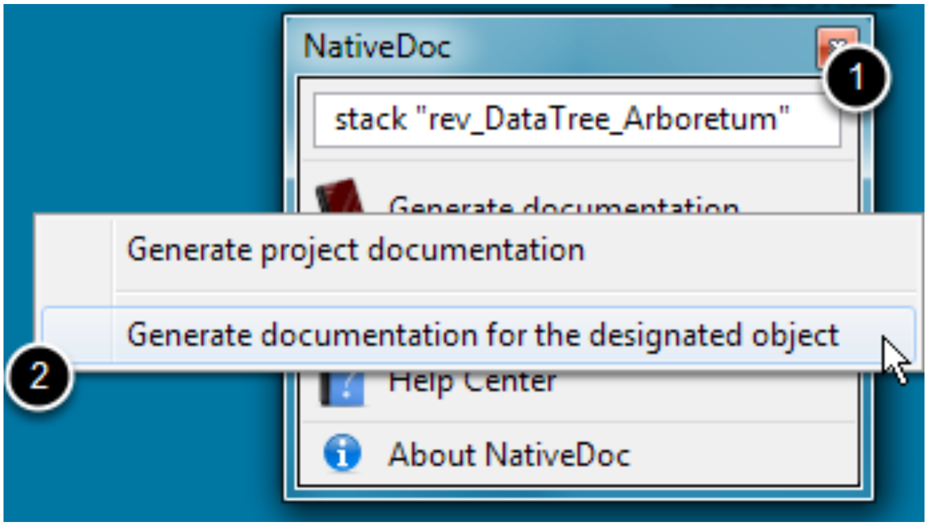
You can refresh the object list by clicking (3) "Refresh object list".

## Second mode: Single object mode

NativeDoc 3.0 has a new documentation generation window, that is more straighforward and more complete.

**The single object generation mode:** Generate the documentation for a single object.

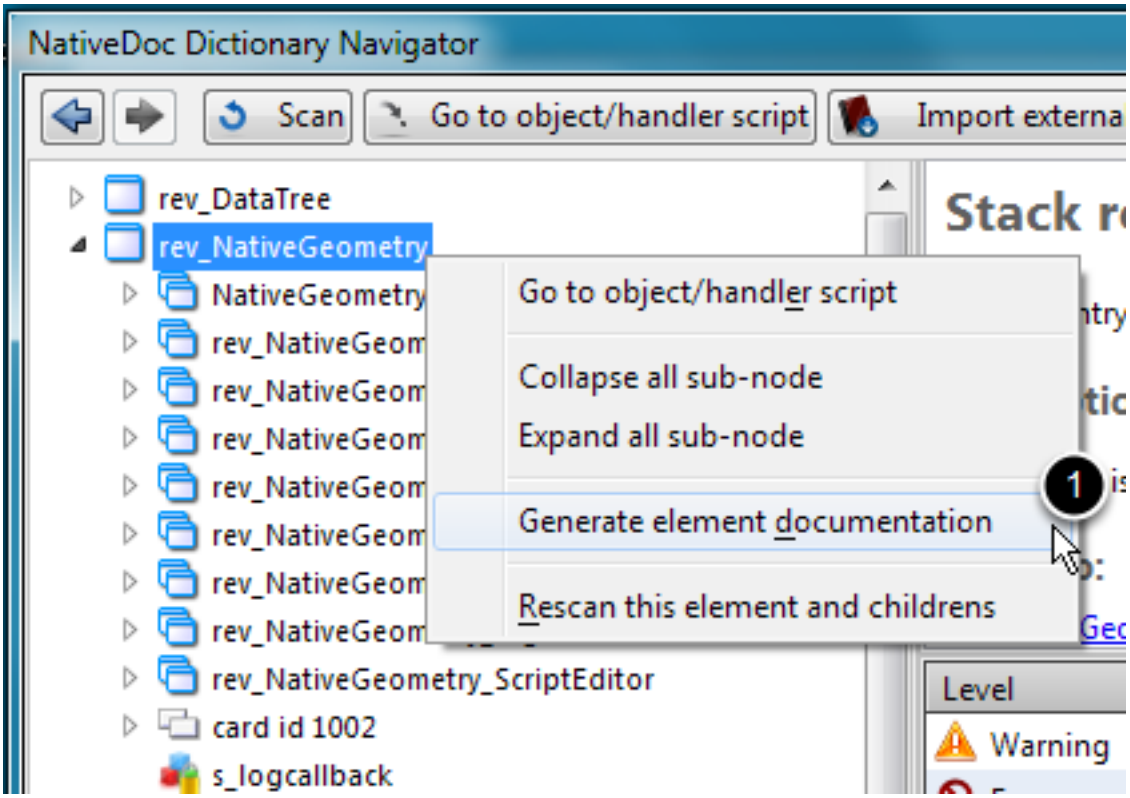
### Generate a documentation for a single object


NativeDoc
stack "rev_DataTree_Arboretum"
1
Generate documentation
Generate project documentation
Generate documentation for the designated object
2
Help Center
About NativeDoc

Select an object or enter manually the object that you want to generate the documentation for in the field (1).

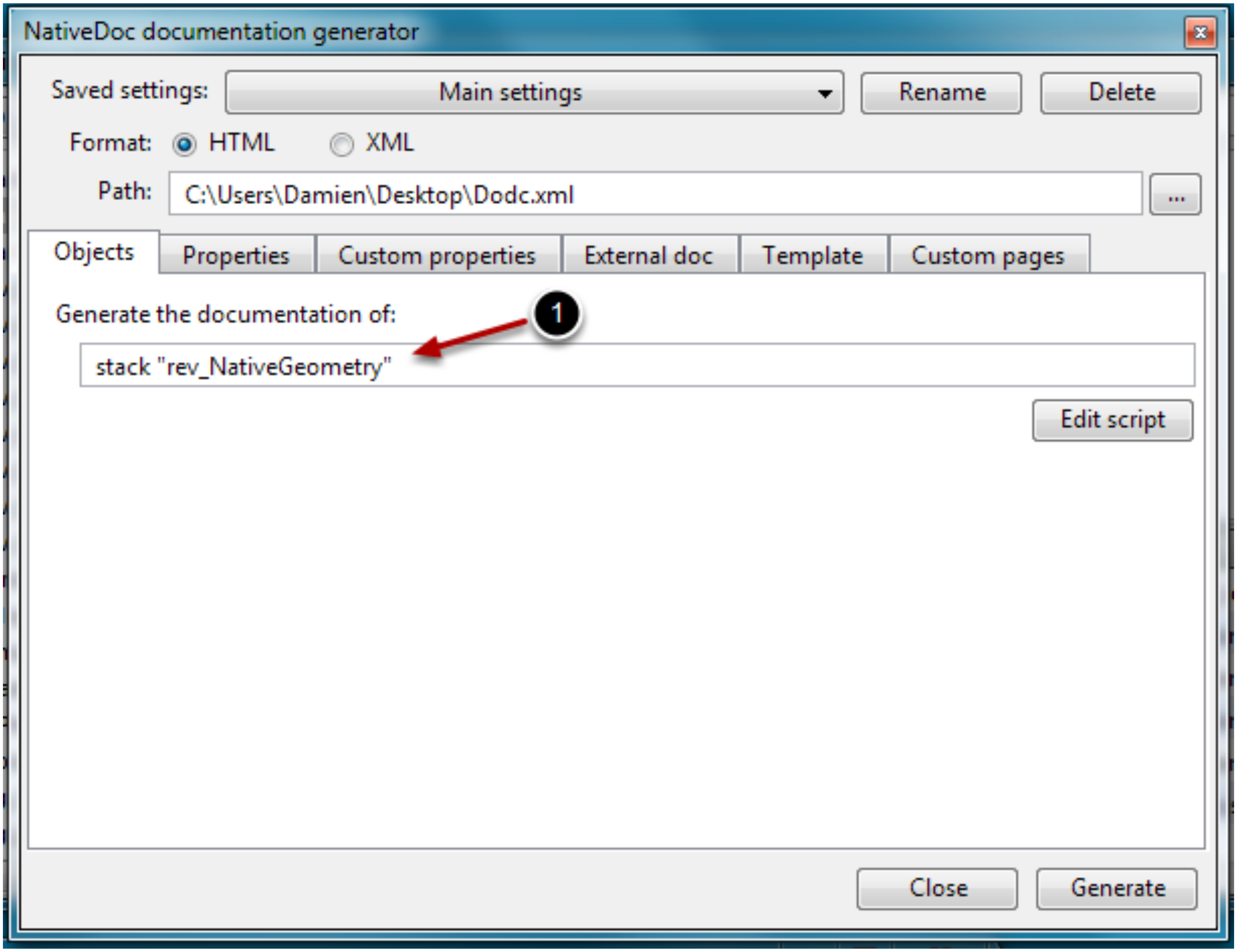
Then click "Generate documentation" and click (2) "Generate documentation for the designated object".

## Generate a documentation for a single object from the Navigator


NativeDoc Dictionary Navigator
Scan
Go to object/handler script
Import externa
rev_DataTree
rev_NativeGeometry
NativeGeometry
rev_NativeGeom
rev_NativeGeometry_ScriptEditor
card id 1002
s_logcallback
Go to object/handler script
Collapse all sub-node
Expand all sub-node
Generate element documentation
Rescan this element and childrens
1
Stack r
Level
Warning

Right click on an element in the Navigator tree and click "Generate element documentation".

## The documentation generator window



NativeDoc documentation generator

Saved settings: Main settings

Format: ● HTML ○ XML

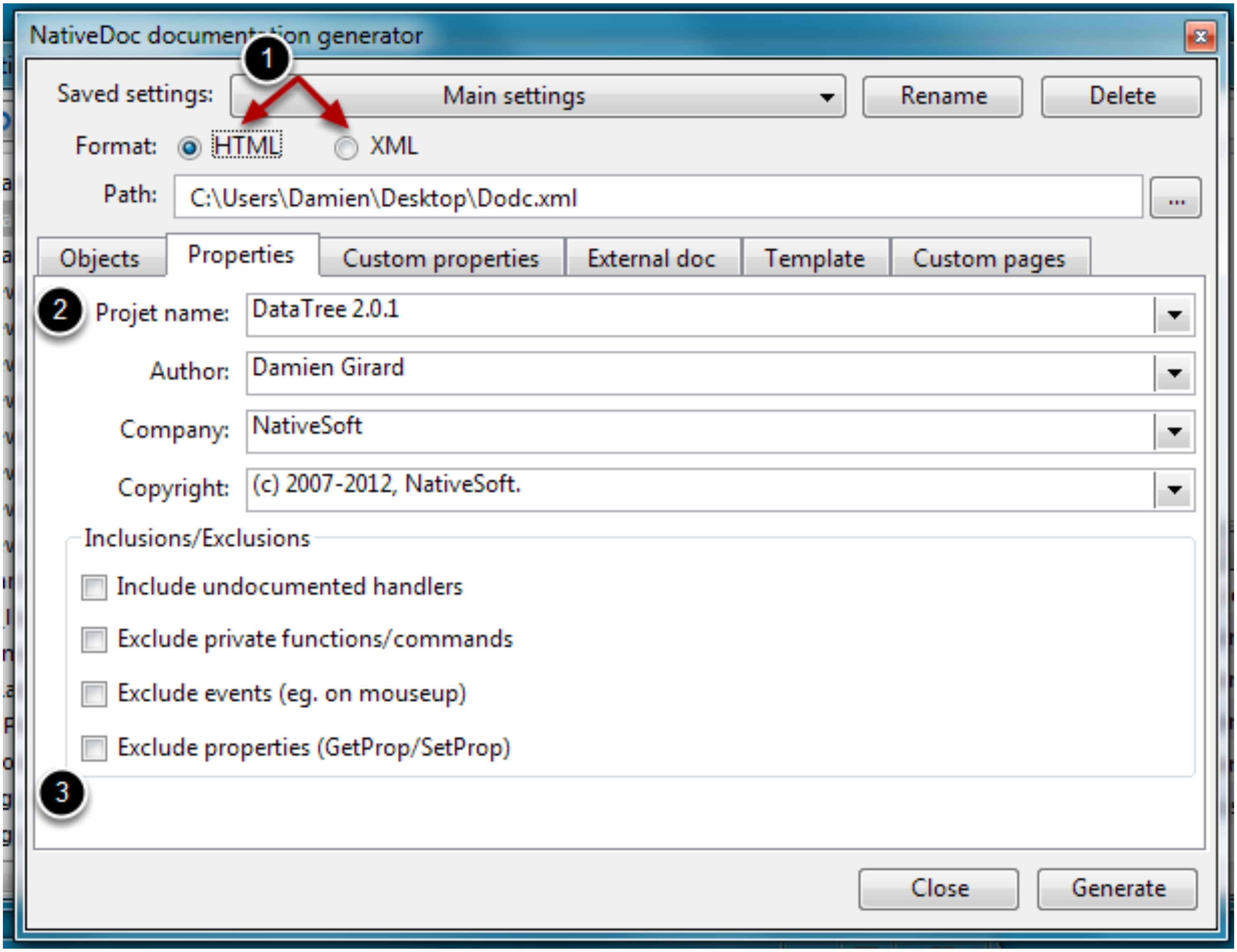
Path: C:\Users\Damien\Desktop\Dodc.xml

Generate the documentation of:

stack "rev_NativeGeometry"

The selected object is displayed in (1).

## General Properties



Saved settings: Main settings

Format: ● HTML ○ XML

Path: C:\Users\Damien\Desktop\Dodc.xml

Projet name: DataTree 2.0.1

Author: Damien Girard

Company: NativeSoft

Copyright: (c) 2007-2012, NativeSoft.

Inclusions/Exclusions

□ Include undocumented handlers

□ Exclude private functions/commands

□ Exclude events (eg. on mouseup)

□ Exclude properties (GetProp/SetProp)

You have first to choose the generation format. (1).

Then you can set the general properties (2), and you can select what you want to include/exclude from the documentation. (3)

**Include undocumented handlers**
Display entirely your application inside the documentation, including handlers not documented.

**Do not include private functions/commands**
Remove all private functions/commands from the documentation.

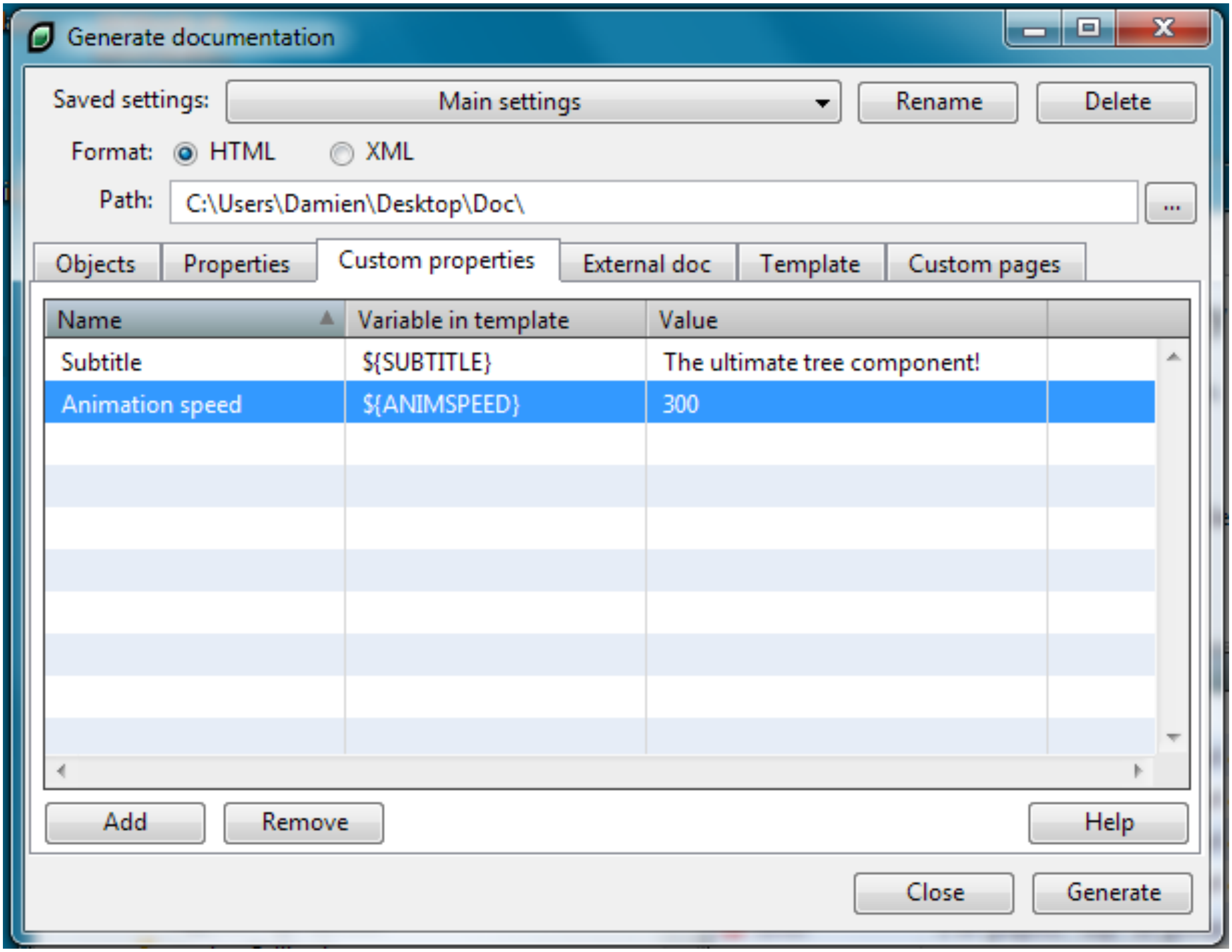
All handlers that have the tag "\private" defined will be excluded too from the documentation.

**Do not include events. (eg. on mouseup)**
Remove all events from the documentation.

**Warning:** An handler is considered as event if it begins with "on" instead of "command"!

**Do not include properties (GetProp/SetProp)**
Remove all getprop/setprop from the documentation.

## Custom properties



Saved settings: Main settings

Format: ● HTML ○ XML

Path: C:\Users\Damien\Desktop\Doc\

| Name | Variable in template | Value |
|---|---|---|
| Subtitle | ${SUBTITLE} | The ultimate tree component! |
| Animation speed | ${ANIMSPEED} | 300 |

In XML generation, those properties will be saved under the "/properties" node.

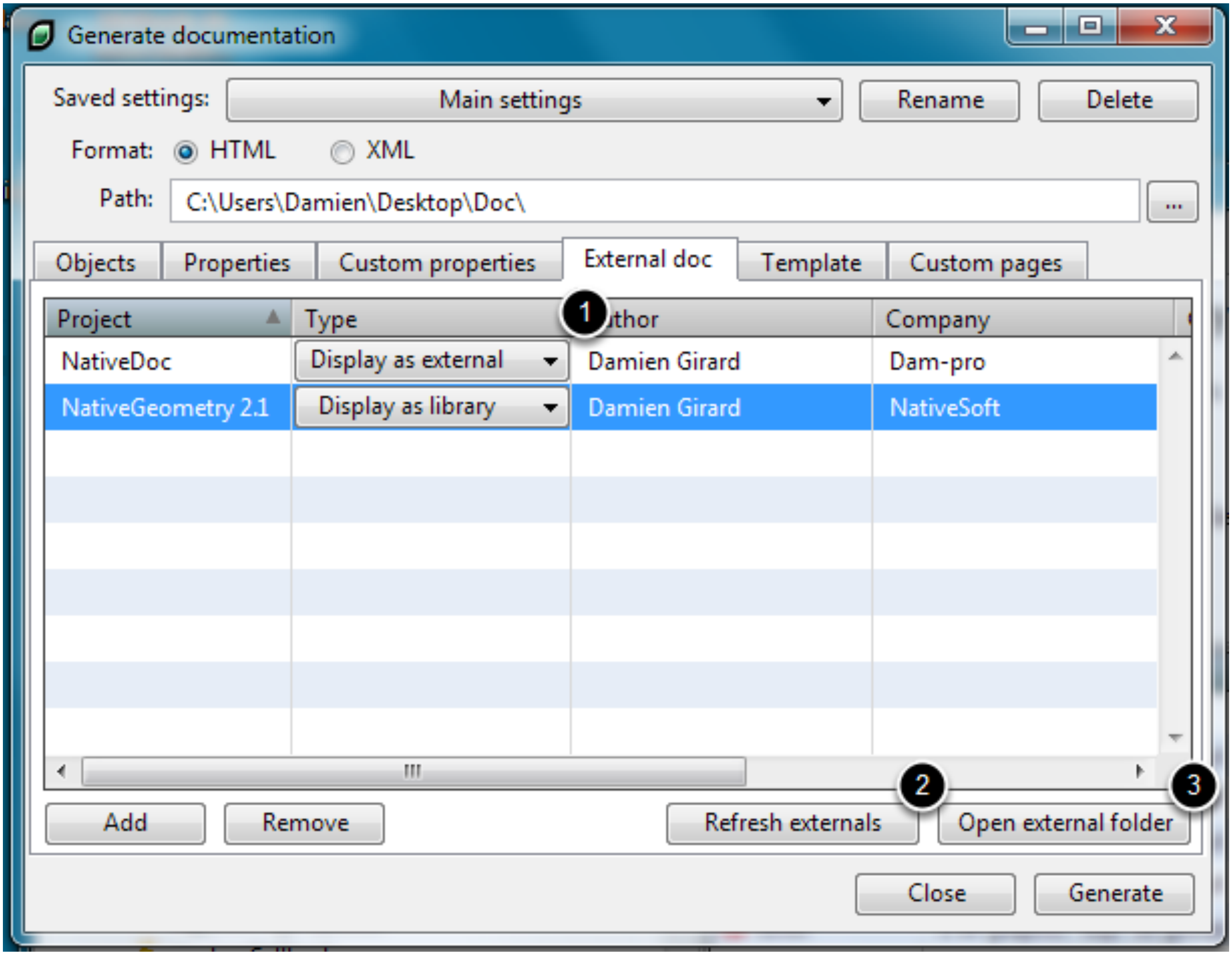
In HTML generation, all files in the template will be scanned and the text in the column "Variable in template" text **will be replaced by the value** defined under the column "Value".

Eg. ${ANIMSPEED} will be replaced by 300 in the template.

It is useful if your custom template has custom properties.

**To edit values**: double click on the element in the grid, and you will be able to edit it.

## External documentation



Generate documentation

Saved settings: Main settings

Format: ● HTML ○ XML

Path: C:\Users\Damien\Desktop\Doc\

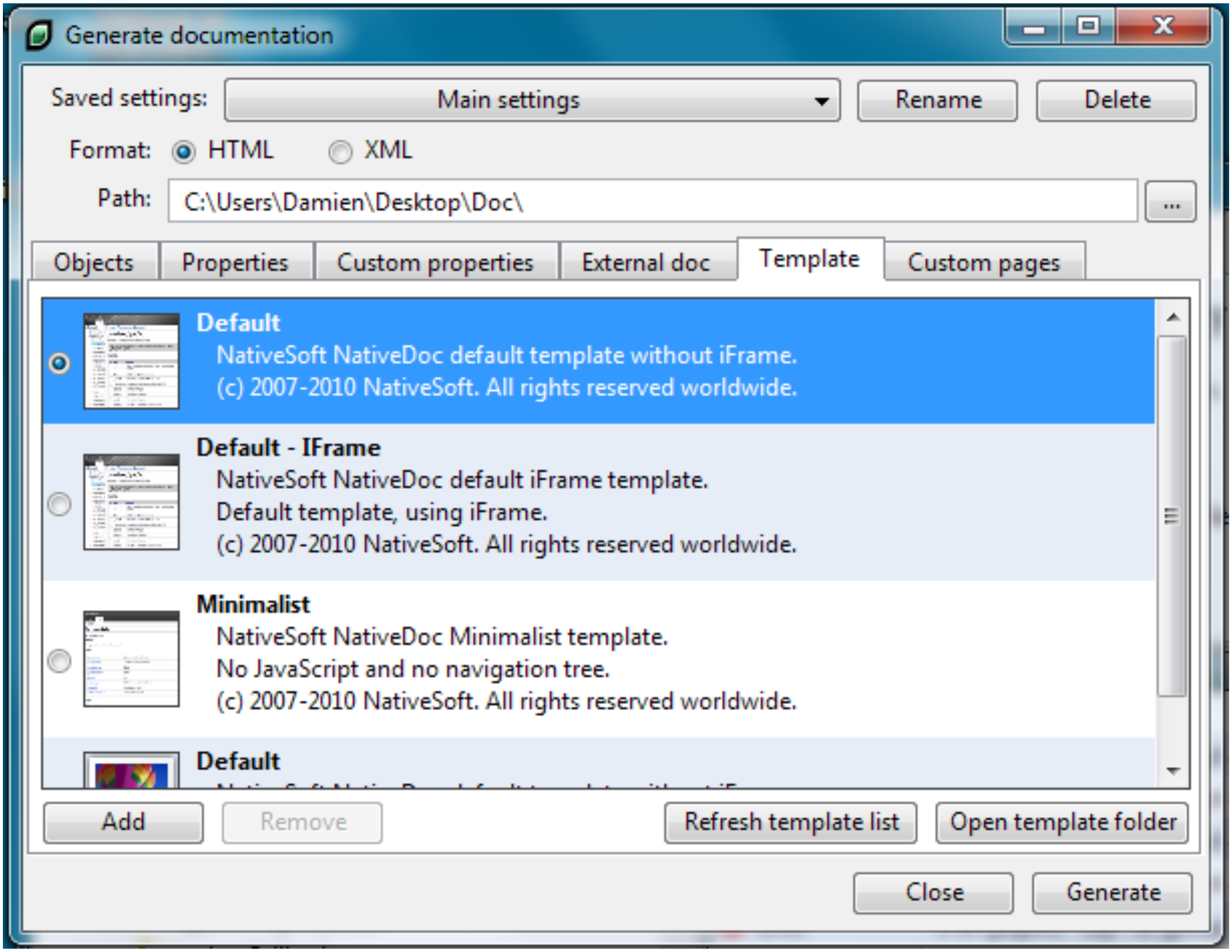
| Project | Type | thor | Company |
|---|---|---|---|
| NativeDoc | Display as external | Damien Girard | Dam-pro |
| NativeGeometry 2.1 | Display as library | Damien Girard | NativeSoft |

You can import several external documentation inside the documentation that you will generate.

In the column (1) "Type" you can choose how to display the documentation inside the generated documentation:
- Integrated: Means that the documentation will not appear as an external.
- Display as library: The documentation will appear under the "Library" section.
- Display as external: The documentation will appear under the "External" section.

You can refresh the external list by clicking (2) "Refresh externals".
To open the external folder, click (3) "Open external folder".

## Template



Saved settings: Main settings

Format: ● HTML ○ XML

Path: C:\Users\Damien\Desktop\Doc\

● **Default**
NativeSoft NativeDoc default template without iFrame.
(c) 2007-2010 NativeSoft. All rights reserved worldwide.

○ **Default - IFrame**
NativeSoft NativeDoc default iFrame template.
Default template, using iFrame.
(c) 2007-2010 NativeSoft. All rights reserved worldwide.

○ **Minimalist**
NativeSoft NativeDoc Minimalist template.
No JavaScript and no navigation tree.
(c) 2007-2010 NativeSoft. All rights reserved worldwide.

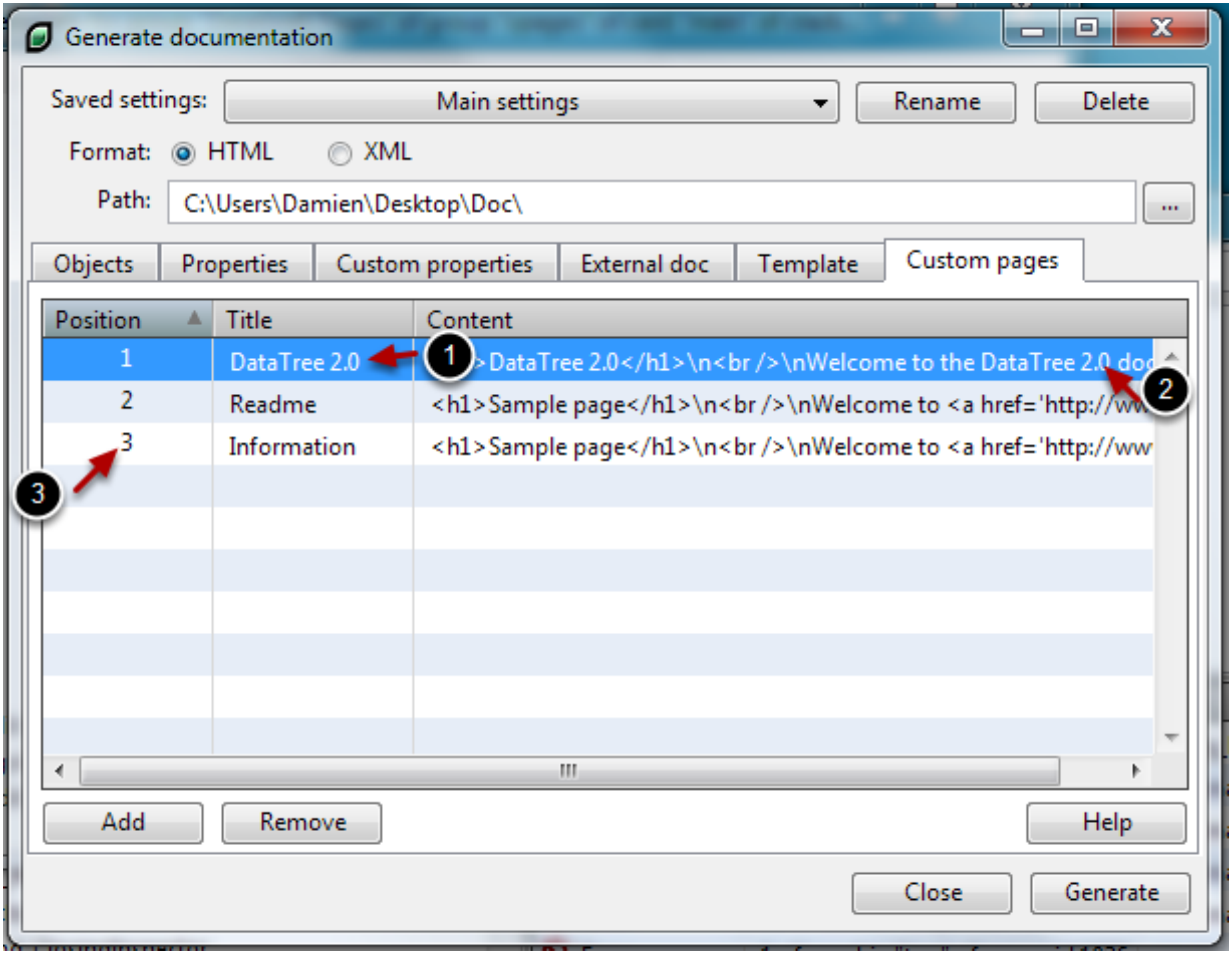
**Default**

Only in HTML generation.

Select which template you want to use in your generation.

**Note:** You can only "Remove" template that you have manually added in this list. (using the "Add" button)

## Custom pages



Generate documentation

Saved settings: Main settings

Format: ● HTML ○ XML

Path: C:\Users\Damien\Desktop\Doc\

| Position | Title | Content |
|---|---|---|
| 1 | DataTree 2.0 | >DataTree 2.0</h1>\n<br />\nWelcome to the DataTree 2.0 doc |
| 2 | Readme | <h1>Sample page</h1>\n<br />\nWelcome to <a href='http://ww |
| 3 | Information | <h1>Sample page</h1>\n<br />\nWelcome to <a href='http://www |

Only in HTML generation.

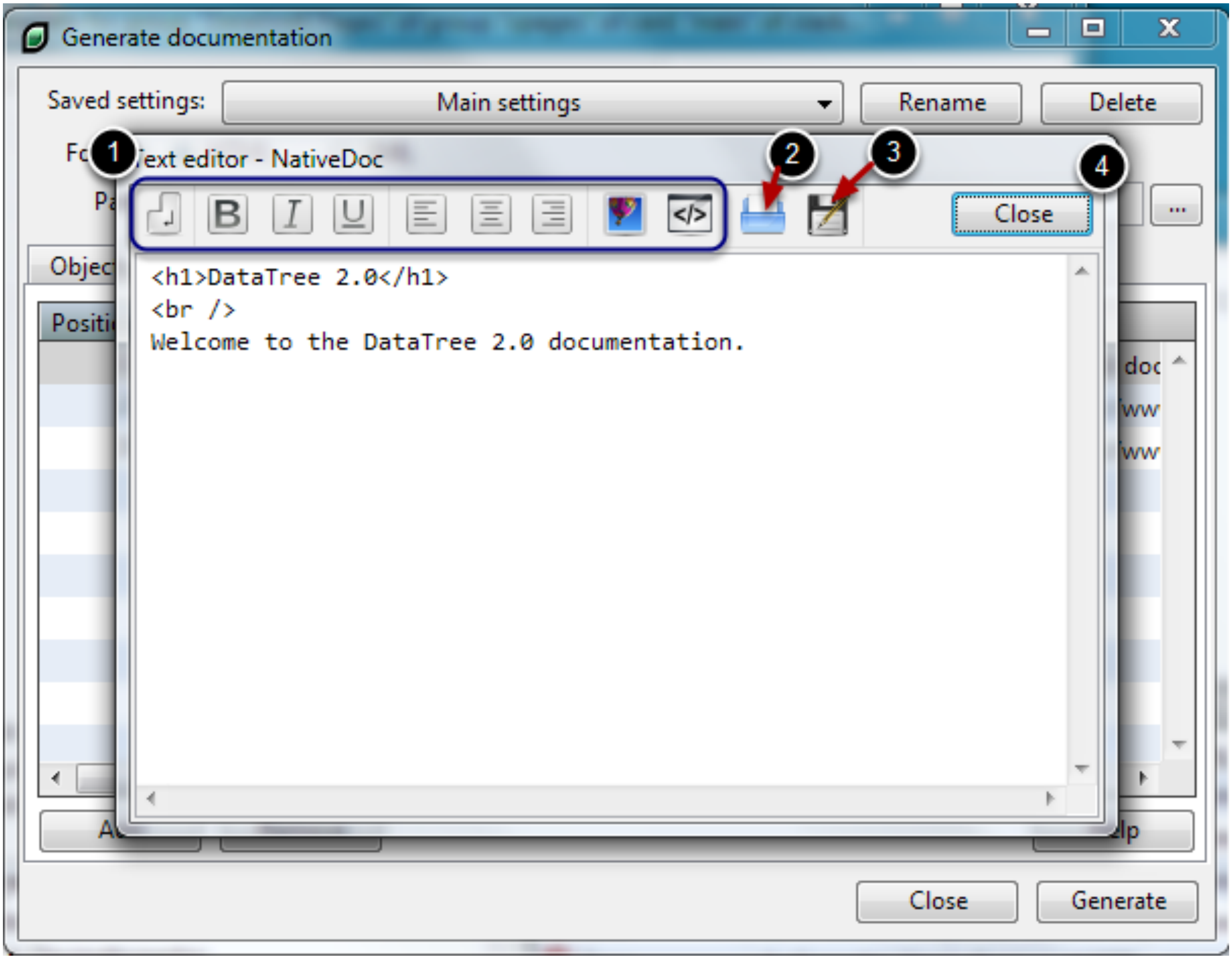
You can add custom pages to your documentation. You can add as many custom page as you want.

To edit the title of the page, double click on the page title (1), to edit the content of the page, double click on the content column (2).

You can set the position of the page by clicking on the first column "Position". (3)

**Note:** You can define the documentation mainpage by setting the position to "Mainpage".

## The custom page editor



```
<h1>DataTree 2.0</h1>
<br />
Welcome to the DataTree 2.0 documentation.
```

The text editor enables you to define your custom page.

**Warning:** The custom page must be in XHTML Strict 1.0 format. If not, web browsers can display your page incorrectly.

Buttons located in (1) helps you to design the page.

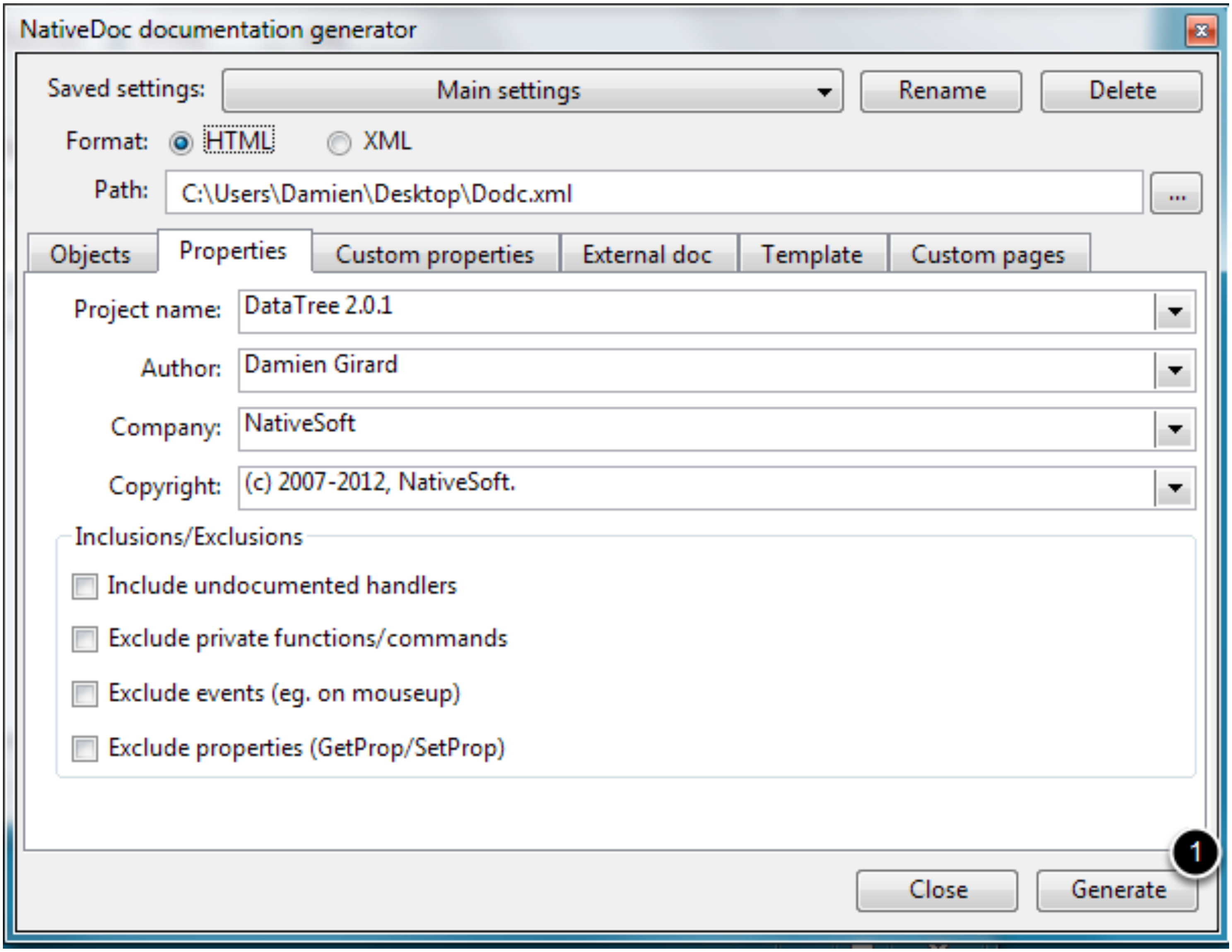
**Open/Save**

To open an external XHTML file, click the open button (2).
To save the edited text, click the save button (3).

**Apply modifications**

To apply modifications, press the "Close" button (4).

## Generating the documentation



Saved settings: Main settings

Format: ● HTML ○ XML

Path: C:\Users\Damien\Desktop\Dodc.xml

Project name: DataTree 2.0.1

Author: Damien Girard

Company: NativeSoft

Copyright: (c) 2007-2012, NativeSoft.

Inclusions/Exclusions

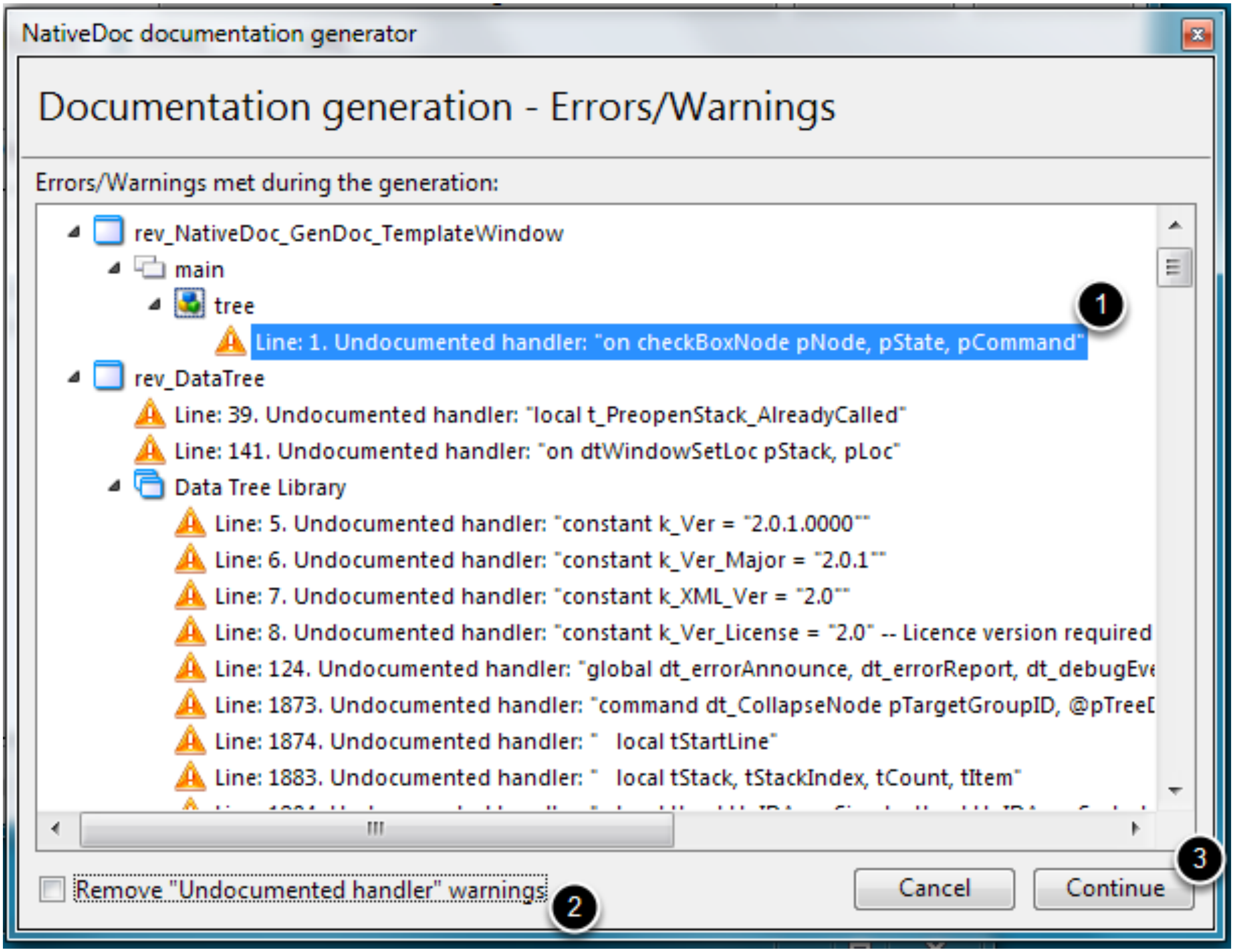
- □ Include undocumented handlers
- □ Exclude private functions/commands
- □ Exclude events (eg. on mouseup)
- □ Exclude properties (GetProp/SetProp)

When you are ready to generate the documentation, click (1) "Generate".

## Documentation generation/exportation error

NativeDoc will shows you all documentation parsing errors.

### The generation error list



NativeDoc documentation generator

Documentation generation - Errors/Warnings

Errors/Warnings met during the generation:

```
rev_NativeDoc_GenDoc_TemplateWindow
    main
        tree
            Line: 1. Undocumented handler: "on checkBoxNode pNode, pState, pCommand"
rev_DataTree
    Line: 39. Undocumented handler: "local t_PreopenStack_AlreadyCalled"
    Line: 141. Undocumented handler: "on dtWindowSetLoc pStack, pLoc"
    Data Tree Library
        Line: 5. Undocumented handler: "constant k_Ver = "2.0.1.0000""
        Line: 6. Undocumented handler: "constant k_Ver_Major = "2.0.1""
        Line: 7. Undocumented handler: "constant k_XML_Ver = "2.0""
        Line: 8. Undocumented handler: "constant k_Ver_License = "2.0" -- Licence version required
        Line: 124. Undocumented handler: "global dt_errorAnnounce, dt_errorReport, dt_debugEve
        Line: 1873. Undocumented handler: "command dt_CollapseNode pTargetGroupID, @pTreeI
        Line: 1874. Undocumented handler: "   local tStartLine"
        Line: 1883. Undocumented handler: "   local tStack, tStackIndex, tCount, tItem"
```

□ Remove "Undocumented handler" warnings

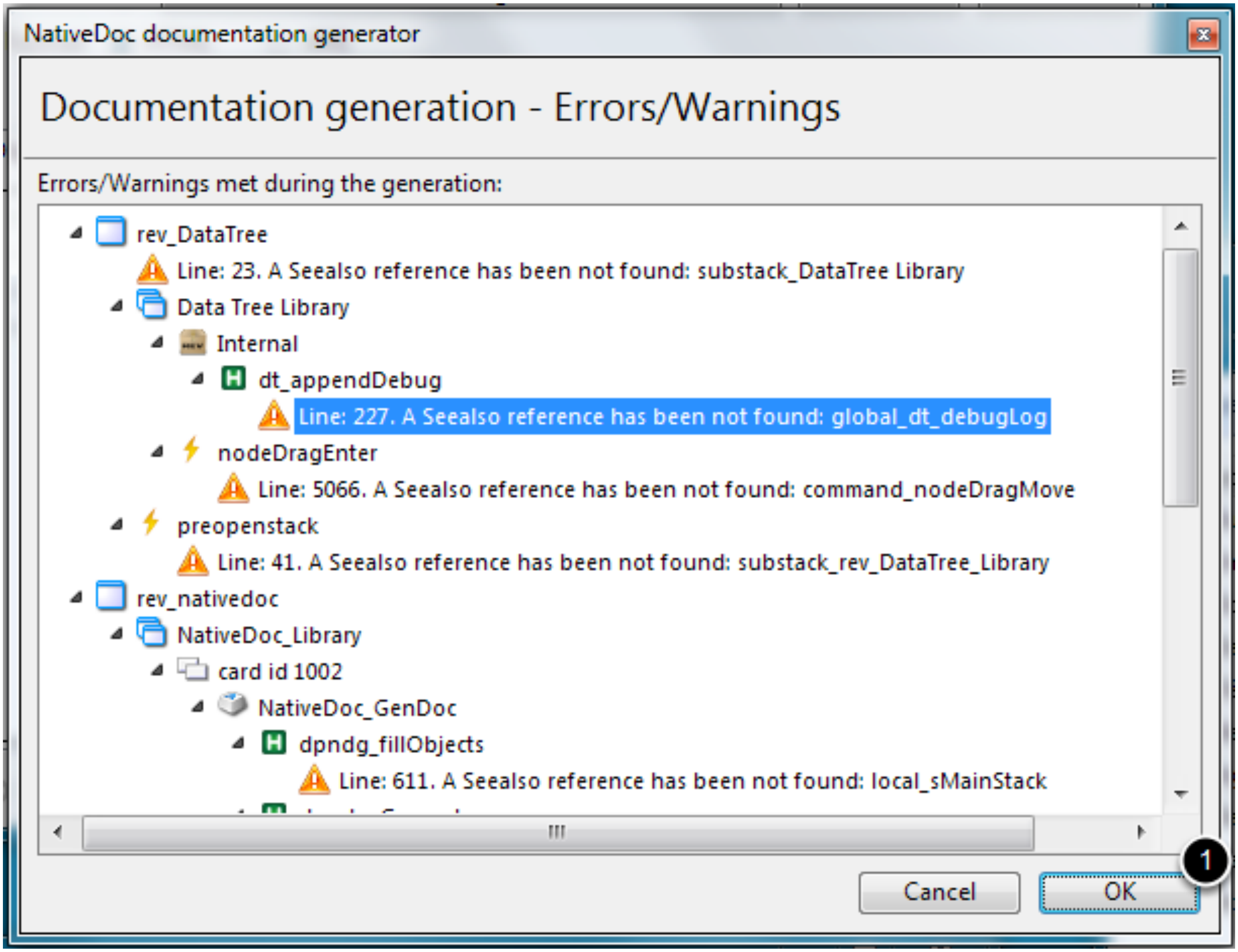
NativeDoc displays all its warning/errors in this tree.

To edit the script of the selected object, double click on the error. (1)

You can filter the list with the "Undocumented handler" warnings. (2)

You can ignore thoses warning/errors and generate the documentation by clicking "Continue" (3).

## The exportation error list



NativeDoc documentation generator
Documentation generation - Errors/Warnings
Errors/Warnings met during the generation:
Cancel
OK
1

```
rev_DataTree
    Line: 23. A Seealso reference has been not found: substack_DataTree Library
    Data Tree Library
        Internal
            dt_appendDebug
                Line: 227. A Seealso reference has been not found: global_dt_debugLog
        nodeDragEnter
            Line: 5066. A Seealso reference has been not found: command_nodeDragMove
    preopenstack
        Line: 41. A Seealso reference has been not found: substack_rev_DataTree_Library
rev_nativedoc
    NativeDoc_Library
        card id 1002
            NativeDoc_GenDoc
                dpndg_fillObjects
                    Line: 611. A Seealso reference has been not found: local_sMainStack
```

Once the documentation exported, NativeDoc displays the exportation errors.

Thoses errors can be file write error, or "seealso" reference not found.
**WARNING:** Oftenly, the givens line is not the line where exactly the error is, but the line of the documentation block or the line of the handler.

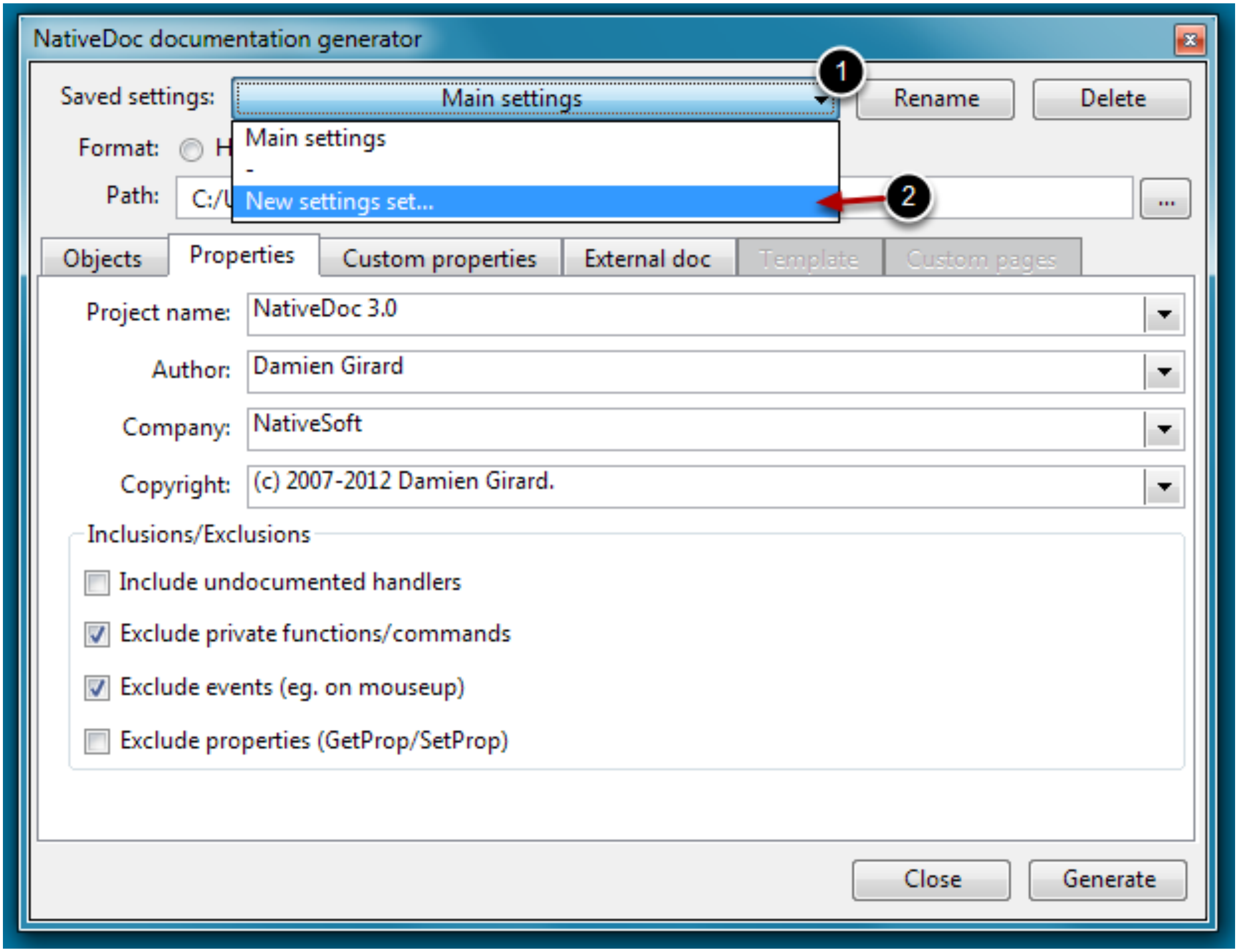
Click "OK" (1) to finish the documentation generation process.

## Using the settings sets

The new generation window allows you to jump between multiple settings set.
Generate several documentations quickly by creating a settings set for each one.

**Note:** A settings set save all settings, **including the objects** selected in project mode.

### Duplicate (Create) a settings set



Saved settings: Main settings
Format: ○
Path: C:/U

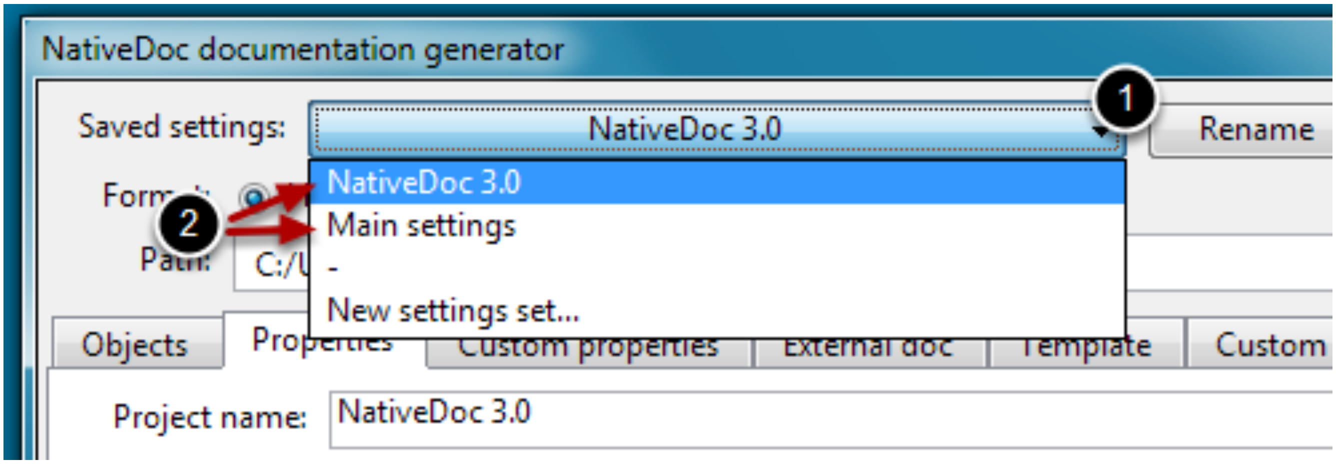
Project name: NativeDoc 3.0
Author: Damien Girard
Company: NativeSoft
Copyright: (c) 2007-2012 Damien Girard.

Inclusions/Exclusions
☐ Include undocumented handlers
☑ Exclude private functions/commands
☑ Exclude events (eg. on mouseup)
☐ Exclude properties (GetProp/SetProp)

Create a new settings set means "Duplicate the current active settings set", so you can copy all settings of a settings set to a new one.

Click the settings set list (1) and then "New settings set..." (2)

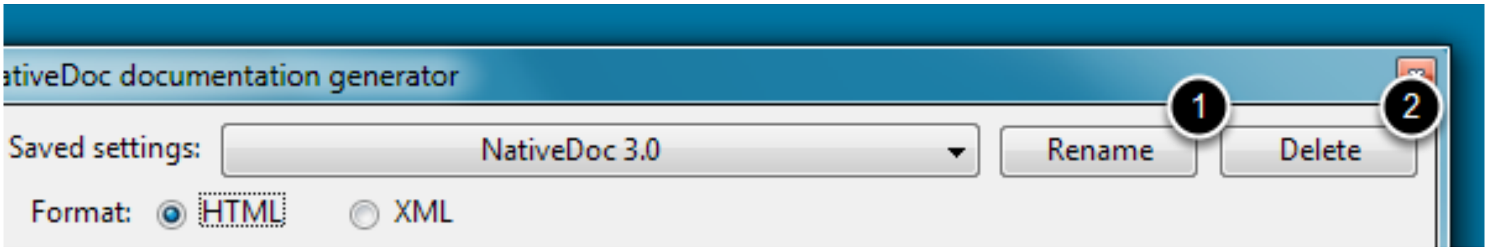
## Jumping from one settings set to another



Click the "Saved settings" menu (1) and choose the settings set that you want to switch to.

**Note:** Settings are saved automatically, when you switch from one settings set to another, all changes are saved.

## Rename/Delete current settings set

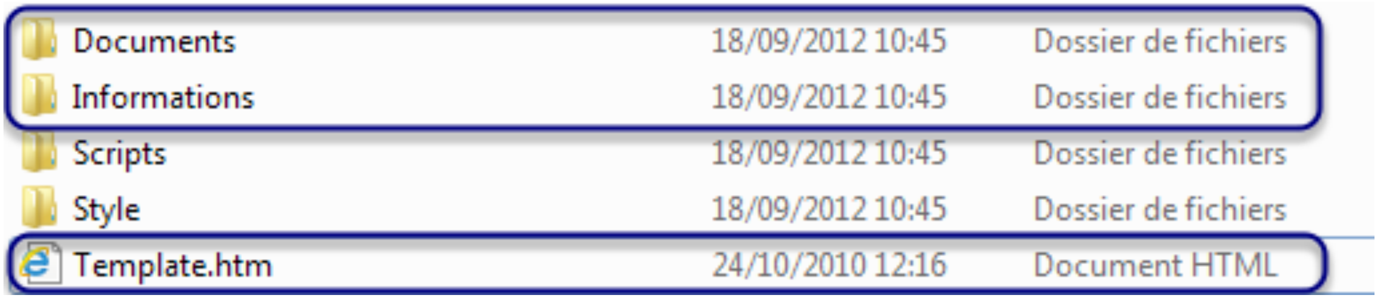


Click (1) "Rename" to rename the active settings set and (2) "Delete" to delete the current settings set.

# How to create a custom template

# Understanding NativeDoc template format

## Structure of a template



| | | |
|---|---|---|
| Documents | 18/09/2012 10:45 | Dossier de fichiers |
| Informations | 18/09/2012 10:45 | Dossier de fichiers |
| Scripts | 18/09/2012 10:45 | Dossier de fichiers |
| Style | 18/09/2012 10:45 | Dossier de fichiers |
| Template.htm | 24/10/2010 12:16 | Document HTML |

A NativeDoc template must have the following files:  
**- /Template.htm**  
**- /Documents/Template.htm**  
**- /Informations/description.txt**

And a preview image as optional:  
**- Informations/preview.png**

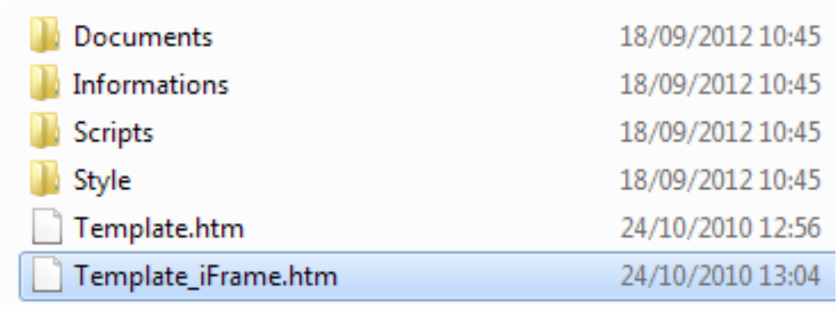
The file /**Template.htm** is used to generate:  
**- The mainpage (index.htm)**  
**- Custom pages**

The file **/Documents/Template.htm** is used to generate the **Documentation pages.**

The file **/Informations/description.htm** is used to display informations inside the NativeDoc documentation generator.

*Note:* Information**s** ends with an "s" because it is written in French. (Same things for Documents).

## Structure of the iFrame template



| | |
|---|---|
| Documents | 18/09/2012 10:45 |
| Informations | 18/09/2012 10:45 |
| Scripts | 18/09/2012 10:45 |
| Style | 18/09/2012 10:45 |
| Template.htm | 24/10/2010 12:56 |
| Template_iFrame.htm | 24/10/2010 13:04 |

The iFrame template has an additionnal file:  
**- Template_iFrame.htm**

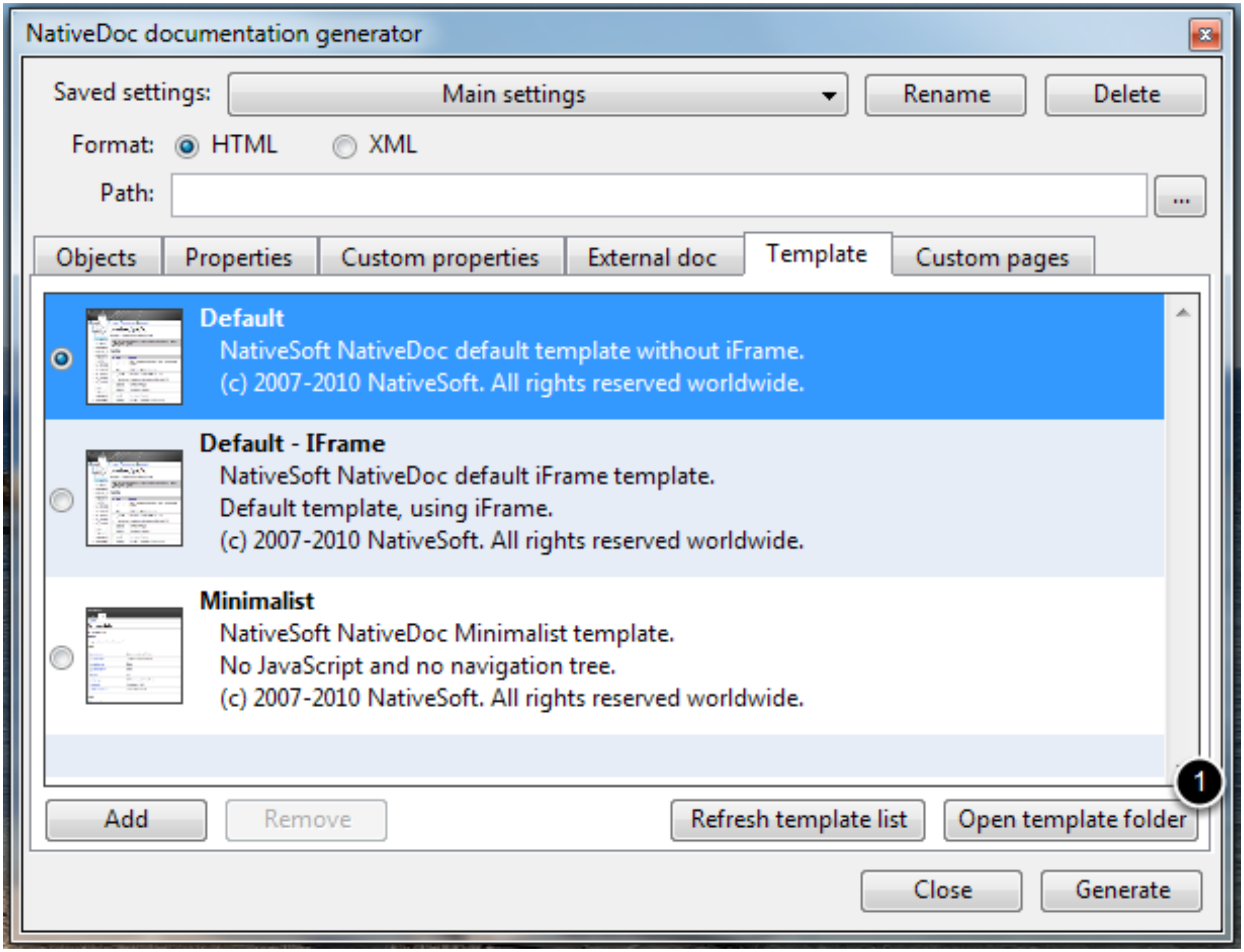
This file will be used to generate the **index.htm** file.

The other files **Template.htm** and **Documents/Template.htm** will contains the body and will be displayed inside the frame.

## Create a template from a NativeSoft template

NativeSoft allows you to modify the default template and default iFrame template in order to create a new template that fit to your needs.

### Creation of a new template



NativeDoc documentation generator

Saved settings: Main settings

Format: ● HTML ○ XML

Path:

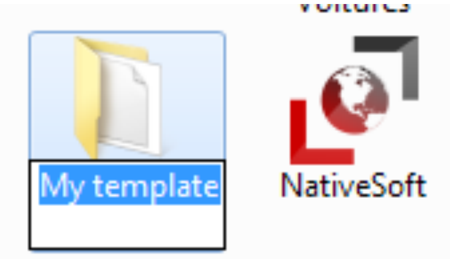
Template

● **Default**
NativeSoft NativeDoc default template without iFrame.
(c) 2007-2010 NativeSoft. All rights reserved worldwide.

○ **Default - IFrame**
NativeSoft NativeDoc default iFrame template.
Default template, using iFrame.
(c) 2007-2010 NativeSoft. All rights reserved worldwide.

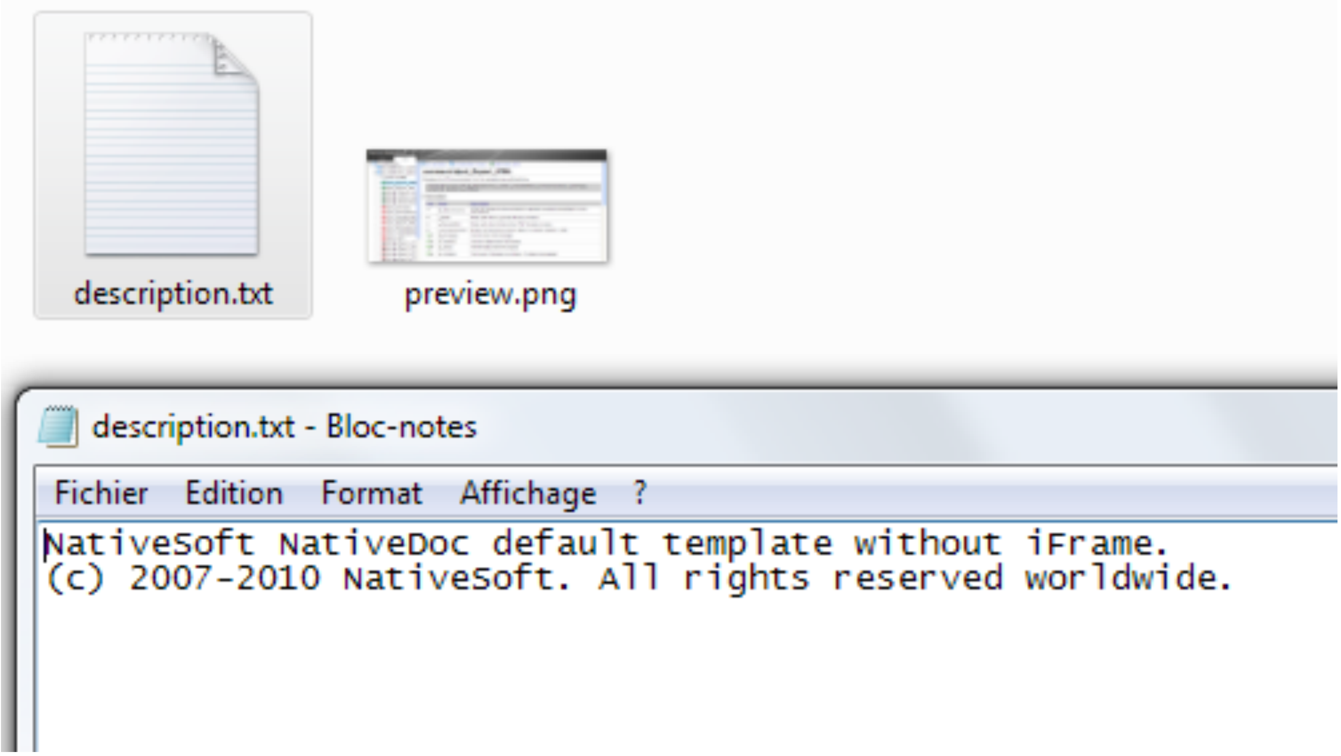
○ **Minimalist**
NativeSoft NativeDoc Minimalist template.
No JavaScript and no navigation tree.
(c) 2007-2010 NativeSoft. All rights reserved worldwide.

Open the default template from the documentation generator by clicking (1) "Open template folder" (1).



Copy the entire folder "Default" to another place and rename it. (eg. My documents)

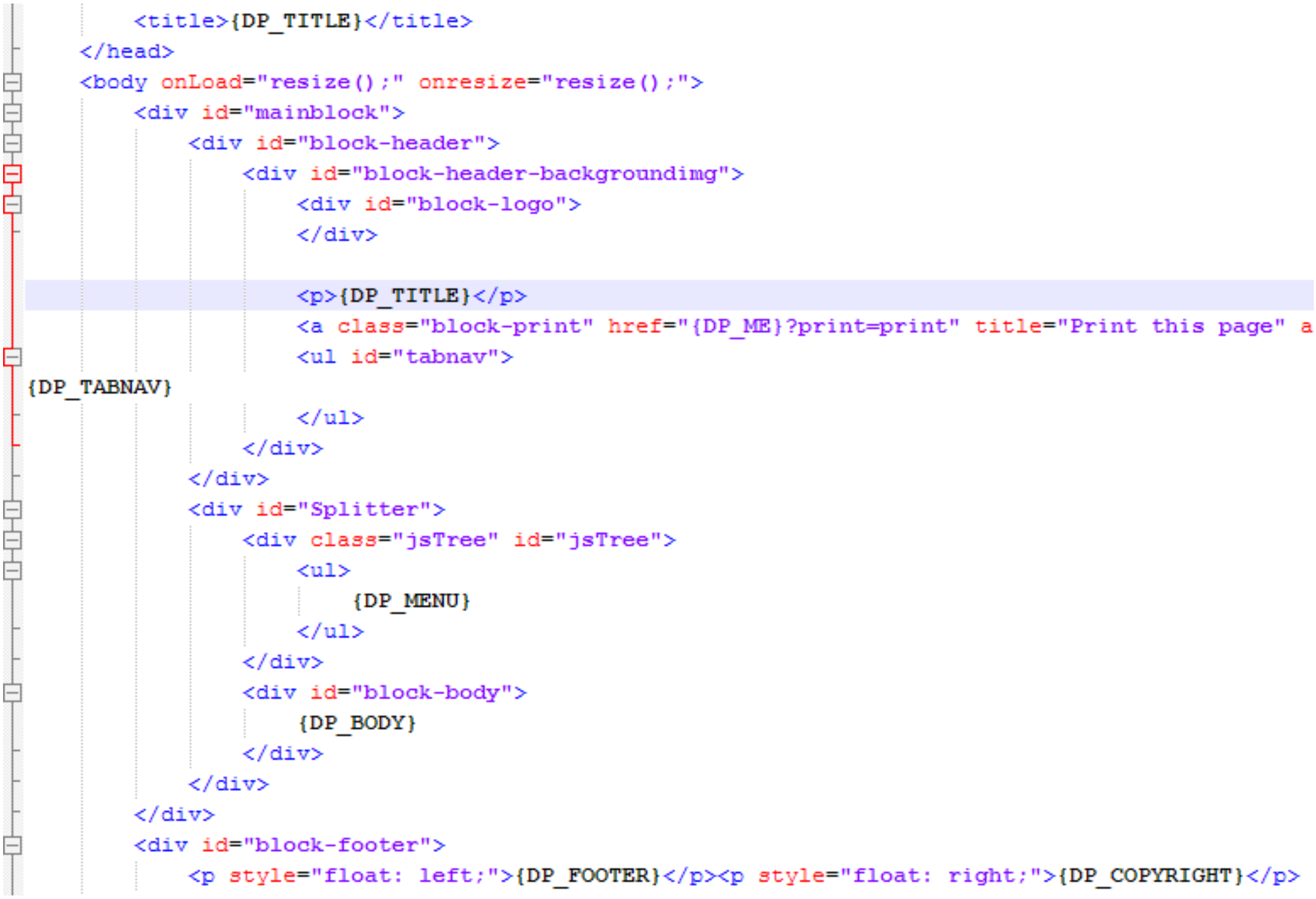
## The description file



```
NativeSoft NativeDoc default template without iFrame.
(c) 2007-2010 NativeSoft. All rights reserved worldwide.
```

Edit the information file located in folder **"Informations/description.txt".**

## Editing the template



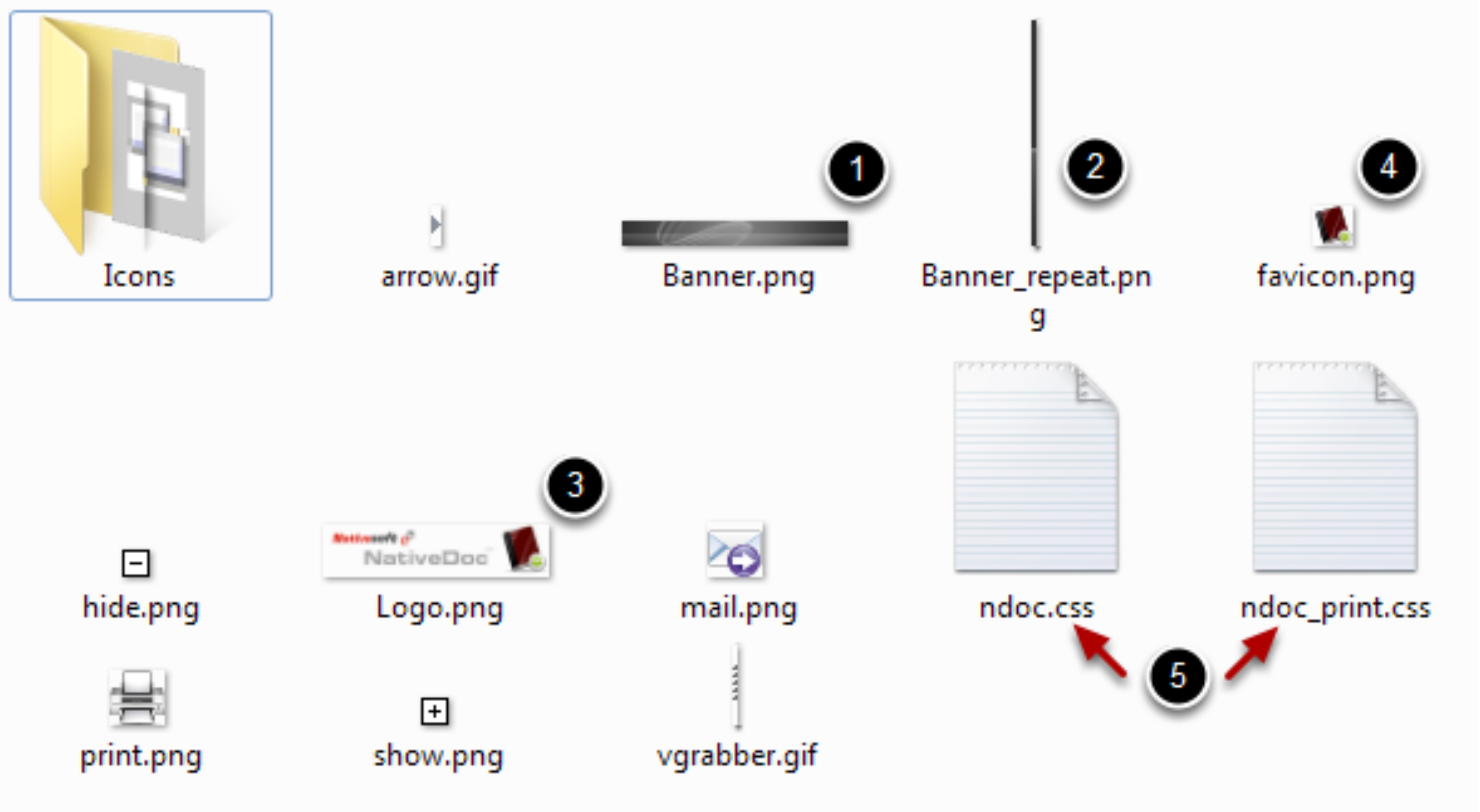
```
        <title>{DP_TITLE}</title>
    </head>
    <body onLoad="resize();" onresize="resize();">
        <div id="mainblock">
            <div id="block-header">
                <div id="block-header-backgroundimg">
                    <div id="block-logo">
                    </div>

                    <p>{DP_TITLE}</p>
                    <a class="block-print" href="{DP_ME}?print=print" title="Print this page" a
                    <ul id="tabnav">
{DP_TABNAV}
                    </ul>
                </div>
            </div>
            <div id="Splitter">
                <div class="jsTree" id="jsTree">
                    <ul>
                        {DP_MENU}
                    </ul>
                </div>
                <div id="block-body">
                    {DP_BODY}
                </div>
            </div>
        </div>
        <div id="block-footer">
            <p style="float: left;">{DP_FOOTER}</p><p style="float: right;">{DP_COPYRIGHT}</p>
```

NativeDoc templates have several variables and the engine will replace them with content:
**- {DP_TITLE}:** Title of the page.
**- {DP_TABNAV}:** Horizontal browser.
**- {DP_MENU}:** JSTree vertical menu. (contains objects tree)
**- {DP_BODY}:** Body
**- {DP_FOOTER}:** Footer
**- {DP_COPYRIGHT}:** Copyright

With custom properties you can add easily your own variable inside the template! See Custom properties.

## Customizing the images and CSS



The file **"Banner_repeat.png"** is the repeated pattern of the banner. To customize the banner, you have to modify those two files:
- **Banner.png** (1)
- **Banner_repeat.png** (3)

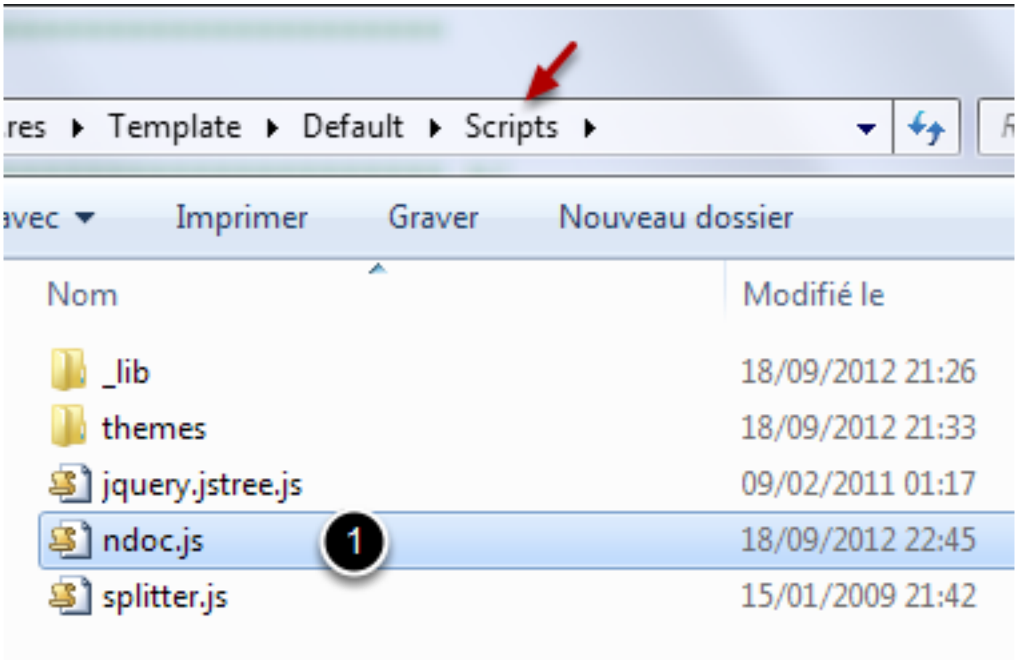
You can replace the file **"Logo.png"** (3) with your company logo and application logo.

The file **"favicon.png"** (4) is the website icon.

The CSS files are:
- **ndoc.css:** Default CSS. (5)
- **ndoc_print.css:** Printer friendly CSS. (5)

If you just want to change the look and feel of the website, just modify those files and you will have a custom template ready to be used.

## Deeper customizations, the script



| Nom | Modifié le |
|---|---|
| _lib | 18/09/2012 21:26 |
| themes | 18/09/2012 21:33 |
| jquery.jstree.js | 09/02/2011 01:17 |
| ndoc.js | 18/09/2012 22:45 |
| splitter.js | 15/01/2009 21:42 |

In the folder **"/Scripts"** is located the file "ndoc.js".

This file contains all the javascript code used to run the template. You can customize this file to create a custom behavior of the generated website.

NativeDoc templates relies on those two libraries:
**- JQuery** - Website: http://www.jquery.com/
**- JSTree -** Website: http://www.jstree.com/

Take a look at those libraries documentation to customize deeply the NativeDoc templates!