

# WebSite X5

## Optional Objects - Developers Guide

# 1. Table of Contents

- [1. Table of Contents](#)
- [2. Syntax and naming conventions](#)
- [3. Optional Objects structure and position of the object folder](#)
  - [3.1. Resources insights](#)
- [4. File localizations.xml](#)
- [5. File manifest.xml](#)
  - [5.1. App Tag](#)
  - [5.2. Name Tag](#)
  - [5.3. Version Tag](#)
  - [5.4. MinimumWebSiteX5Version Tag](#)
  - [5.5. Category Tag](#)
  - [5.6. Description Tag](#)
  - [5.7. PageExtension Tag](#)
  - [5.8. CustomHeight Tag](#)
  - [5.9. Parameters Tag](#)
    - [5.9.1. Fields Tag](#)
      - [5.9.1.1. Text](#)
        - [5.9.1.1.1. XML code supported in manifest](#)
        - [5.9.1.1.2. Properties accessible via JavaScript code](#)
      - [5.9.1.2. Number](#)
        - [5.9.1.2.1. XML code supported in manifest](#)
        - [5.9.1.2.2. Properties accessible via JavaScript code](#)
      - [5.9.1.3. Color](#)
        - [5.9.1.3.1. XML code supported in manifest](#)
        - [5.9.1.3.2. Properties accessible via JavaScript code](#)
      - [5.9.1.4. Colors](#)
        - [5.9.1.4.1. XML code supported in manifest](#)
        - [5.9.1.4.2. Properties accessible via JavaScript code](#)
      - [5.9.1.5. BorderWidth](#)
        - [5.9.1.5.1. XML code supported in manifest](#)
        - [5.9.1.5.2. Properties accessible via JavaScript code](#)
      - [5.9.1.6. RoundCorners](#)
        - [5.9.1.6.1. XML code supported in manifest](#)
        - [5.9.1.6.2. Properties accessible via JavaScript](#)
      - [5.9.1.7. Shadow](#)
        - [5.9.1.7.1. XML code supported in manifest](#)
        - [5.9.1.7.2. Properties accessible via JavaScript](#)
      - [5.9.1.8. Dimensions](#)
        - [5.9.1.8.1. XML code supported in manifest](#)
        - [5.9.1.8.2. Properties accessible via JavaScript](#)
      - [5.9.1.9. Margins](#)
        - [5.9.1.9.1. XML code supported in manifest](#)
        - [5.9.1.9.2. Properties accessible via JavaScript](#)

#### [5.9.1.10. Checkbox](#)

[5.9.1.10.1. XML code supported in manifest](#)

[5.9.1.10.2. Properties accessible via JavaScript](#)

#### [5.9.1.11. Separator](#)

[5.9.1.11.1. XML code supported in manifest](#)

#### [5.9.1.12. Dropdown](#)

[5.9.1.12.1. XML code supported in manifest](#)

[5.9.1.12.2. Properties accessible via JavaScript](#)

#### [5.9.1.13. Font](#)

[5.9.1.13.1. XML code supported in manifest](#)

[5.9.1.13.2. Properties accessible via JavaScript](#)

#### [5.9.1.14. FileList](#)

[5.9.1.14.1. XML code supported in manifest](#)

[5.9.1.14.2. Properties accessible via JavaScript](#)

#### [5.9.1.15. StringList](#)

[5.9.1.15.1. XML code supported in manifest](#)

[5.9.1.15.2. Properties accessible via JavaScript](#)

#### [5.9.1.16. File](#)

[5.9.1.16.1. XML code supported in manifest](#)

[5.9.1.16.2. Properties accessible via JavaScript](#)

#### [5.9.1.17. Link](#)

[5.9.1.17.1. XML code supported in manifest](#)

[5.9.1.17.2. Methods accessible via JavaScript code](#)

#### [5.10. Resources Tag](#)

#### [5.11. ShowPreview Tag](#)

#### [5.12. Hooks Tag](#)

[5.12.1. GetHeight\(\)](#)

[5.12.2. IsEmpty\(\)](#)

[5.12.3. ShowPreview\(\)](#)

[5.12.4. GetHeaderContents\(type, currentContent\)](#)

[5.12.5. OnBeforeFileElaboration\(fieldID, file\)](#)

[5.12.6. OnBeforeImageElaboration\(fieldID, image\)](#)

[5.12.7. OnBeforeResourceElaboration\(resourceID, resource\)](#)

#### [5.13. Tag Output](#)

#### [5.14. Tag PreviewOutput](#)

#### [5.15. Tag UIPreviewOutput](#)

### [6. WSX5Script](#)

### [7. Useful advices about how to write the code](#)

## 2. Syntax and naming conventions

All the XML tag and attribute names are case sensitive.

All the XML files must be correctly indented to provide a better readability.

### 3. Optional Objects structure and position of the object folder

An Optional Object is made of some different files. All of them are stored in a folder which has as name the ID of the Optional Object.

This folder needs to be created in the following path:

```
C:\Users\[User Name]\AppData\Local\Incomedia\WebSiteX5 v11 – Professional\PluginApps
```

An Optional Object folder may contain the following files:

- manifest.xml: View relevant paragraph
- localizations.xml: View relevant paragraph
- icon.png: The icon shown to the user (needs to be 32px\*32px)
- key: file containing the signature of the application (generated by Incomedia)
- Resources: the folder where are stored the Object's resource files. It is possible to define subfolders too.

#### 3.1. Resources insights

The resources are files that are somewhat related to the Optional Object. They may be javascript libraries needed by the Object's output code, example images used in the Object's preview or whatever file the author need to run his Optional Object.

A resource file will be always copied in the output site by default. The author may override this behavior following the instructions provided in the ["Resources Tag"](#) paragraph

### 4. File localizations.xml

The file format is the same used for the languages of WebSite X5.

It is obligatory to fill in at least the English localization in the file.

To access the localizations from the manifest.xml file, add the l10n-id attribute to the xml element you want to localize.

The l10n-id attribute must contain a localization id that was previously specified in the localization.xml file using the <Key> tag as reported in the following example.

Example structure:

```
<?xml version="1.0" encoding="utf-8"?>
<WebsiteX5LanguagesLibrary xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Languages>
    <Language>
      <Code><![CDATA[EN]]></Code>
      <Name><![CDATA[English]]></Name>
      <WritingDirection><![CDATA[LeftToRight]]></WritingDirection>
      <Localizations>
```

```
<Localization>  
<Key>L10n Key</Key>  
<Value>Localization value</Value>  
</Localization>  
</Localizations>  
</Language>  
</Languages>  
</WebsiteX5LanguagesLibrary>
```

## 5. File manifest.xml

Describes the Object structure and features and contains the links to the resource files, the Object's code, and the user interface descriptors. Here you can find a list of the tags it may contain.

### 5.1. App Tag

Mandatory. It is the opening tag. It has the attribute `uuid` which identifies the app uniquely.

```
<App uuid=" UNIQUEID ">
```

### 5.2. Name Tag

Mandatory. It contains the name of the application.

```
<Name> Name of the app </Name>
```

### 5.3. Version Tag

Mandatory. It contains a integer to identify the version of the application.

```
<Version> Version of the application </Version>
```

### 5.4. MinimumWebSiteX5Version Tag

Optional. It contains the minimum required version of WebSiteX5 to run the optional object.

### 5.5. Category Tag

Mandatory. It contains the ID of the category where the app will be available.

```
<Category>Category of the Application</Category>
```

### 5.6. Description Tag

Mandatory. It contains the description of the application. It can be localized with the `l10n-id` and `global-l10n-id` properties.

<Description I10n-id="description">Description of the Application</Description>

### 5.7. PageExtension Tag

Optional. If present it contains the extension of the page required to allow the code of the Object to work correctly. The extension has to be added without characters like “\*.”. For example, to require a php extension it is only necessary to write:

```
<PageExtension>php</PageExtension>
```

### 5.8. CustomHeight Tag

Optional. If provided and has an integer numerical value higher than zero, it indicates the height of the object. If provided, the automatic height calculation will be deactivated.

```
<CustomHeight>100</CustomHeight>
```

### 5.9. Parameters Tag

Optional. It contains a list of “Tab” tags which represents every Tab shown in the Object’s UI.

Each Tab tag may have an id attribute and may contain the following subtags:

- **Label:** Set the tab’s label shown in the UI. It may have the I10n-id attribute for localization purposes.
- **ShowPreview:** if set to “true” shows the preview in the selected tab.
- **Fields:** See the following paragraph

#### 5.9.1. Fields Tag

The Field tag must contains a list of Field tag which are the input fields visible to the user in the UI of WebSite X5.

Each Field tag needs to have an unique id defined using the “id” attribute.

Every “Field” tag can contain:

- The “**type**” attribute: which defines the type of field according to the list you will find below.
- a **Label** subtag: defines the label shown near the field. This tag can have the attribute “position” with as values “top”, “left” or “none”.
- a **Mandatory** tag: set to true to set the field as mandatory
- The **UpdatesPreview** subtag which, if set as true, will update automatically the preview with every change in the UI. As default it is set as false.
- A **Hooks** subtag where following JS functions can be defined (without need of the use of the tags <?wsx5 ?>):
  - **OnCreate():** Is called before generating the field. The output of this function will be used to generate the field contents. Note: at the moment only the dropdown field actually uses this function.
  - **OnValueChanged():** Is called when the user changes the value of this field in the UI of WebSite X5.

- The **Description** subtag which can be localized via the attributes l10n-id and global-l10n-id, with the description of the field. It can have the attribute “style” where the value can be “normal” (default) or “info”.
- An **Indent** subtag with a numerical integer value bigger or equal to zero. It indicates the indent level of the field according to the left border of the form.
- A **Position** subtag with the word “auto” or an integer value bigger or equal to zero. In case it contains “auto” the fields will be positioned automatically. In case it contains an integer value the field will be positioned in the same position of the field indicated by the value.

Here you can find a list of the available field types.

The values represented in the following XML code examples are not to consider as default values for the various fields.

#### 5.9.1.1. Text

Show a text input.

##### 5.9.1.1.1. XML code supported in manifest

```
<Field type="text" id="">
  <DefaultValue>DefaultValue</DefaultValue>
  <IsPassword>false</IsPassword>
  <ShowPasswordCreation>false</ShowPasswordCreation>
  <MaxLength>10</MaxLength>
  <Multiline>false</Multiline>
  <LinesCount>1</LinesCount>
  <ShowScrollBar>false</ShowScrollBar>
  <Label l10n-id="loc_id">Default label text</Label>
</Field>
```

##### 5.9.1.1.2. Properties accessible via JavaScript code

- value (string)

#### 5.9.1.2. Number

Allows to choose a number

##### 5.9.1.2.1. XML code supported in manifest

```
<Field type="number" id="">
  <DefaultValue>10</DefaultValue>
  <MinValue>0</MinValue>
  <MaxValue>100</MaxValue>
  <Increment>5</Increment>
  <ShowDecimals>false</ShowDecimals>
  <Label l10n-id="loc_id">Default label text</Label>
</Field>
```

#### 5.9.1.2.2. Properties accessible via JavaScript code

- value (number)

#### 5.9.1.3. Color

Allows to choose a color from a palette.

##### 5.9.1.3.1. XML code supported in manifest

```
<Field type="color" id="">
  <EnableTransparent>>false</EnableTransparent>
  <DefaultValue>(#AARRGGBB|#RRGGBB)</DefaultValue>
  <Label l10n-id="loc_id">Default label text</Label>
</Field>
```

#### 5.9.1.3.2. Properties accessible via JavaScript code

- value (hexadecimal string)
- valueR (integer 0-255)
- valueG (integer 0-255)
- valueB (integer 0-255)
- valueA (integer 0-255)

#### 5.9.1.4. Colors

Allows to choose four colors.

##### 5.9.1.4.1. XML code supported in manifest

```
<Field type="colors" id="">
  <EnableTransparent>>false</EnableTransparent>
  <DefaultValue>(#AARRGGBB|#RRGGBB)</DefaultValue>
  <Label l10n-id="loc_id">Default label text</Label>
</Field>
```

#### 5.9.1.4.2. Properties accessible via JavaScript code

- top (hexadecimal string)
- bottom (hexadecimal string)
- left (hexadecimal string)
- right (hexadecimal string)
- leftR (integer 0-255)
- leftG (integer 0-255)
- leftB (integer 0-255)
- leftA (integer 0-255)
- rightR (integer 0-255)
- rightG (integer 0-255)
- rightB (integer 0-255)
- rightA (integer 0-255)
- topR (integer 0-255)



- topG (integer 0-255)
- topB (integer 0-255)
- topA (integer 0-255)
- bottomR (integer 0-255)
- bottomG (integer 0-255)
- bottomB (integer 0-255)
- bottomA (integer 0-255)

#### 5.9.1.5. BorderWidth

Allows to choose the width of 4 borders.

##### 5.9.1.5.1. XML code supported in manifest

```
<Field type="borderwidth" id="">
  <DefaultValue>1</DefaultValue>
  <MinValue>0</MinValue>
  <MaxValue>100</MaxValue>
  <Increment>5</Increment>
  <Label l10n-id="loc_id">Default label text</Label>
</Field>
```

##### 5.9.1.5.2. Properties accessible via JavaScript code

- left (integer)
- right (integer)
- top (integer)
- bottom (integer)

#### 5.9.1.6. RoundCorners

Allows to choose the rounding of the corners.

##### 5.9.1.6.1. XML code supported in manifest

```
<Field type="roundcorners" id="">
  <DefaultValue>1</DefaultValue>
  <MinValue>0</MinValue>
  <MaxValue>100</MaxValue>
  <Increment>5</Increment>
  <Label l10n-id="loc_id">Default label text</Label>
</Field>
```

##### 5.9.1.6.2. Properties accessible via JavaScript

- topLeft (integer)
- topRight (integer)
- bottomLeft (integer)
- bottomRight (integer)

### 5.9.1.7. Shadow

Allows to choose the style of a shadow.

#### 5.9.1.7.1. XML code supported in manifest

```
<Field type="shadow" id="">
  <DefaultEnabled>>false</DefaultEnabled>
  <DefaultColor>(#AARRGGBB|#RRGGBB)</DefaultColor>
  <DefaultOffsetX>1</DefaultOffsetX>
  <DefaultOffsetY>1</DefaultOffsetY>
  <DefaultDimension>10</DefaultDimension>
  <DefaultDiffusion>10</DefaultDiffusion>
  <Label l10n-id="loc_id">Default label text</Label>
</Field>
```

#### 5.9.1.7.2. Properties accessible via JavaScript

- enabled
- color (hexadecimal string)
- colorR (integer 0-255)
- colorG (integer 0-255)
- colorB (integer 0-255)
- colorA (integer 0-255)
- diffusion (integer)
- dimension (integer)

### 5.9.1.8. Dimensions

Allows to choose width and height using a single UI control.

#### 5.9.1.8.1. XML code supported in manifest

```
<Field type="dimensions" id="">
  <DefaultValue>100</DefaultValue>
  <MinValue>0</MinValue>
  <MaxValue>100</MaxValue>
  <Increment>5</Increment>
  <KeepRatio>>true</KeepRatio>
  <Label l10n-id="loc_id">Default label text</Label>
</Field>
```

#### 5.9.1.8.2. Properties accessible via JavaScript

- width
- height

### 5.9.1.9. Margins

Allows to choose 4 margins.

#### 5.9.1.9.1. XML code supported in manifest

```
<Field type="margins" id="">
  <DefaultValue>1</DefaultValue>
  <MinValue>0</MinValue>
  <MaxValue>100</MaxValue>
  <Increment>5</Increment>
  <Label l10n-id="loc_id">Default label text</Label>
</Field>
```

#### 5.9.1.9.2. Properties accessible via JavaScript

- top (integer)
- bottom (integer)
- left (integer)
- right (integer)

### 5.9.1.10. Checkbox

Allows to create a check box.

#### 5.9.1.10.1. XML code supported in manifest

```
<Field type="checkbox" id="">
  <DefaultChecked>true</DefaultChecked>
  <Label l10n-id="loc_id">Default label text</Label>
</Field>
```

#### 5.9.1.10.2. Properties accessible via JavaScript

- checked (Boolean)

### 5.9.1.11. Separator

Allows to add a separator. A separator can change the behavior of a panel (column) inside the UI of the application.

It can have a sub tag `NewPanel` which, if it contains "true" this forces the creation of a new section in a new panel.

#### 5.9.1.11.1. XML code supported in manifest

```
<Field type="separator" id="">
  <Label l10n-id="loc_id">Default label text</Label>
  <NewPanel>true</NewPanel>
</Field>
```

### 5.9.1.12. Dropdown

It is a field which is represented like a drop down menu.

If a class is defined with the attribute “class” then it will be filled automatically by webSite X5 based on the class value.

the available class values are the following:

- database: shows a dropdown menu to select a database defined in step 4

#### 5.9.1.12.1. XML code supported in manifest

```
<Field type="dropdown" id="" shownumbers="(true|false)"
class="database">
  <Options>
    <Option
      l10n-id="localization_id"
      value="field_value">
        Default text
    </Option>
  </Options>
  <Hooks><![CDATA[
function OnCreate() {
  // Must return the options array.
  // Each option element is defined via a JSON object
  // containing a text and a value property.
  return [{ text: "option text", value: "option value" }];
}
]]></Hooks>
  <DefaultValue>value_default_option</DefaultValue>
  <Label l10n-id="loc_id">Default label text</Label>
</Field>
```

#### 5.9.1.12.2. Properties accessible via JavaScript

- value (string)

### 5.9.1.13. Font

It is a field which allows to choose the font, it's size and style.

#### 5.9.1.13.1. XML code supported in manifest

```
<Field type="font" id="">
  <DefaultFontFamily>Tahoma</DefaultFontFamily>
  <DefaultFontSize>10</DefaultFontSize>
  <DefaultBold>>false</DefaultBold>
  <DefaultItalic>>false</DefaultItalic>
  <Label l10n-id="loc_id">Default label text</Label>
</Field>
```

#### 5.9.1.13.2. Properties accessible via JavaScript

- family (string)
- size (integer pt)
- bold (Boolean)
- italic (Boolean)

#### 5.9.1.14. FileList

Allows to choose a list of files.

##### 5.9.1.14.1. XML code supported in manifest

```
<Field type="filelist" id="">
  <Extensions>jpg,txt</Extensions>
  <Label l10n-id="loc_id">Default label text</Label>
</Field>
```

#### 5.9.1.14.2. Properties accessible via JavaScript

- list (array of FileSelect elements)
  - list.length (number of elements in the array)
  - list.path (path for the file example list[0].path prints the path of the first file)

#### 5.9.1.15. StringList

Allows to add a list of strings.

##### 5.9.1.15.1. XML code supported in manifest

```
<Field type="stringlist" id="">
  <Label l10n-id="loc_id">Default label text</Label>
</Field>
```

#### 5.9.1.15.2. Properties accessible via JavaScript

- list (strings array)

#### 5.9.1.16. File

Allows to choose a file.

##### 5.9.1.16.1. XML code supported in manifest

```
<Field type="file" id="">
  <Extensions>jpg,txt</Extensions>
  <Label l10n-id="loc_id">Default label text</Label>
</Field>
```

#### 5.9.1.16.2. Properties accessible via JavaScript

- path (string)
- name (string)
- extension
- isImage (Boolean)

- width: only if isImage == true
- height: only if isImage == true
- clones: only if isImage == true, contains the list of clones of the image. Everyone has the fields shown above.

#### 5.9.1.17. Link

Allows to choose a link from the link selection window of WebSite X5.

##### 5.9.1.17.1. XML code supported in manifest

```
<Field type="link" id="">  
  <Label l10n-id="loc_id">Default label text</Label>  
</Field>
```

##### 5.9.1.17.2. Methods accessible via JavaScript code

- getHTML(string content)

## 5.10. Resources Tag

Optional. If present, it contains a list of tag Resource. Each of them defines the operations to perform on the files available in the “Resources” folder of the application.

The tag can have following properties:

- **id**: optional, an unique ID assigned to the resource
- **src**: mandatory, the source path of the file relative to the “Resources” folder of the application. You cannot provide a path outside that folder, that is, provide a path that contains the “../” string.
- **action**: optional, may contain “copy” or “process”. The default property is “copy”. If “copy”, the file will be copied as is by WebSite X5. If “process”, the content of the file will be elaborated according the same rules used for the content in the Output tag (see below)
- **autolink**: optional. If the linked file is a js or css file and this field is set to “true”, the file will be automatically linked in the header of the website page. The default value is “false”.
- **shared**: optional. If set to “true” the resource will be copied only once in a shared folder for all objects of this type. This will allow to avoid duplicated copies of a static file. The default value is “false”.
- **offlineonly**: the resource will be copied or processed only in the offline mode (UI preview and site preview).

## 5.11. ShowPreview Tag

Optional. If set as “true”, the UI of WebSite X5 will show a preview of the object directly in the window where the input fields are available.

## 5.12. Hooks Tag

Optional. This tag contains the WSX5Script code which has to be run in specific moments in the creation of the object. The script can define following functions.

### 5.12.1. GetHeight()

If the tag CustomHeight has not been set in the XML code of manifest, this function will be called every time it will be necessary to calculate the height of the object. It has to return an integer value greater than zero.

If the CustomHeight tag has not been set in the manifest and this hook is not present, WebSite X5 will use its own height algorithm, which can be less precise.

### 5.12.2. IsEmpty()

If present, this hook has to return a boolean value. It has to return true if the object has to be considered empty, false otherwise.

### 5.12.3. ShowPreview()

Available from Version 11.0.4. If present, this hook has to return a boolean value. It returns the value true if the object is able to show the preview in the UI of the software on contrary false.

### 5.12.4. GetHeaderContents(type, currentContent)

This function allows the application author to specify a custom code to include in the head tag of the website.

This function has to return a string, which will be included in the best position according to the “type” parameter.

The type parameter indicated the required type of content and can have following string values:

- js
- css
- other

The currentContent parameter is a read-only string that contains the current headers for the type of code specified via the “type” parameter. It is useful to verify to not add a duplicated code.

This function can be called more times during the creation of the page code (more precisely, one time for every possible value of the “type” parameter).

To avoid the duplication of the code, the programmer has to always verify the value of the “type” parameter and act consequently.

### 5.12.5. OnBeforeFileElaboration(fieldID, file)

This function is called every time the object elaborates a generic file selected by the user except images (for images see the next hook).

- **fieldID**: contains the ID of the form field which has been used by the final user for the file selection.
- **file**: contains the json object which allows to access to the file data and to perform commands on it. Has the following elements:
  - **name**,
  - **extension**,
  - **isCreationPrevented**,
  - **preventCreation()**
  - **content**: allows to get and set the content of the file. In this way it will be possible to modify the content of the files selected by the user

### 5.12.6. OnBeforeImageElaboration(fieldID, image)

This function is called every time the object elaborates an image file selected by the user.

- **fieldID**: Contains the ID of the form field which has been used by the user to select the file.
- **image**: Contains the json object which allows to access the the image data and to perform some operations of it. In addition to the attributes present in the file parameter of the OnBeforeFileElaboration function, the parameter image is a json file which contains following items:
  - **sourceWidth**,
  - **sourceHeight**,
  - **outputWidth**,
  - **outputHeight**,
  - **clone(filename, width, height)**: creates a clone of the image with different dimensions.

### 5.12.7. OnBeforeResourceElaboration(resourceID, resource)

Available from Version 11.0.4.

This function is called everytime the object elaborates a resource file linked in the manifest.

- **resourceID**: contains the ID of the resource
- **resource**: contains the json object which allow to access to the data of the resource and to perform some operations on it. Has the following elements:
  - **name**: the original name of the file
  - **extension**
  - **destination**: the path where the file will be saved
  - **isCreationPrevented**
  - **preventCreation()**



### 5.13. Tag Output

Mandatory. This tag contains the code which has to be used to create the HTML code to insert inside the page of the website.

Inside this tag it can be added any type of code. It will be then copied inside the code of the page.

The code WSX5Script must be executed within the `<?wsx5` and `?>` tag. This code follows the guide lines defined in the WSX5Script paragraph.

### 5.14. Tag PreviewOutput

Optional. If available, it contains the output HTML code to show in the preview. Works like the Output tag.

### 5.15. Tag UIPreviewOutput

Optional. If available, it contains the output HTML code to show in the small preview visible in the UI. Works like the Output tag.

## 6. WSX5Script

The WSX5Script code is an ECMAScript code which is run by WebSite X5 when the HTML/CSS code is built. The script output will be inserted where the script is.

The syntax to use is the same as JavaScript.

The script needs to be executed inside the `<?wsx5 ?>` tags (excpetion of the Hooks tag which do not require any particular tag).

Following global variables are accessible through code (further informations available on <http://help.websitex5.com/api/pluginapps/>):

- **parameters:** each one has the properties indicated in the “Tag Parameters” paragraph. Further, everyone of them has these global methods and properties:
  - **hide():** hides the field in the UI.
  - **show():** shows the field in the UI.
  - **visible:** true if the field is visible in the UI.
  - **enable():** abilitates the field in the UI.
  - **disable():** disables the field in the UI.
  - **enabled:** true if the field is enabled in the UI.
- **resources:** An object containing, for every ID, an object with following properties:
  - **id:** the ID specified in the manifest
  - **name:** name of the file
  - **destination:** path relative to the root of the website
- **document:** has following methods:
  - **write(string):** writes a string in output
- **console**

- log(string): add a message in the developers console (tab visible when the app is in developer mode).
- **currentObject**: has following properties:
  - id: id of the object
  - width: width of the object
- **I10n**: Allows you to read a value of the localization.xml file and has following methods:
  - **get(localizationId)**: returns the localization in the website language.
  - **get(localizationId, defaultValue)**: returns the localization in the website language. If localizationId is not found, returns defaultValue.
  - **get(localizationId, defaultValue, languageld)**: returns the localization in the specified language, returning defaultValue id localizationId is not found.
  - **get\_global(localizationId)**: returns the requested localization by taking it from the localization library of the UI of WebSite X5. It is always in the installation language of the software.
  - **get\_global(localizationId, DefaultValue)**: returns the requested localization by taking it from the localization library of the UI of WebSite X5. It is always in the installation language of the software. In case the localization is not present it will be used the default value.  
As example, to have as output a value you added in the localization.xml you can use `<?wsx5 document.write(I10n.get("Localization Key")) ?>` where instead of Localization Key you add the Key used in the localization.xml file.
- **wsx5**: has following properties
  - **uiLanguageld**: User Interface Language
  - **contentLanguageld**: Site Language
  - **mode**: 'uipreview', 'preview', 'online', according to the moment when the content is made
  - **accessManagement**: informations about the private area
  - **shoppingCart**: informations on the shopping cart
  - **dataManagement**: informations on the projects data

## 7. Useful advices about how to write the code

If the algorithm to calculate automatically the height is used (in other words the tag CustomHeight and the hook GetHeight have not been used), to allow to calculate the correct height of the cell, it is strongly recommended to set the css height and width properties in the images included within the object.