Lighting Analysts
illumination engineering software

FlashTools Documentation

Last Updated: 23/Aug/2017

Thank you for purchasing the lighting FlashTools offered by Lighting Analysts, Inc. This document contains all the information describing the XML interface your webmaster needs to get up and running with the FlashTools. The XML interface allows you to integrate technologies you are already using (ASP, ASP.NET, PHP, Perl, etc.) to send XML data between your server and the FlashTools.

Getting Started

You received a starter kit (ZIP file) containing the following files. You should also have a separate SWF that is built specifically for your site.
You will need to customize all of these files (except the SWF) to suit your website.

`index.html`	A sample webpage which links to the `FlashTool.htm` webpage
`FlashTool.htm`	A sample webpage which embeds the FlashTool movie (SWF).
`FlashTool.swf`	The FlashTool movie. Note: You will need to replace the default SWF with the SWF built for your site.
`FlashTools_Input.xml`	A sample XML input file for the FlashTool.
`[IES_Files]`	A folder containing sample IES files.
`[Images]`	A folder containing sample images for the logo, background, luminaires, and custom command buttons.
`SaveXML.php`	A sample script which saves the XML output of the FlashTool.
`PhotometricDB.php`	A sample script which accepts an IES file name as input and returns its file path and or data.

Note*: The above scripts will only work if your server supports PHP. The above scripts are very primitive and* ***will not work*** *for your website. They must be replaced and supplemented if you want additional (server-side) functionality.*

(1) Copy the above files to a folder on your web server. *http://www.YourDomain.com/YourFolder/*
(2) Go to `index.html` in your web browser. *http://www.YourDomain.com/YourFolder/index.html*
(3) Click on the appropriate link to load a working version of the FlashTool.

If you experience problems, make sure the folder on your web server has appropriate permissions.

Part A -- Open the FlashTool in a web page

Sample Files: index.html, FlashTool.htm, FlashTool.swf

index.html is the webpage which contains information about luminaires. On this page next to each product there can be a link (or equivalent) to start the FlashTool. These links are composed of two parts.

(1) JavaScript function (or equivalent) in the document HEAD to open the new window.

```
<script language="JavaScript" type="text/JavaScript">
function openFlashTool_XML() {
     var A = "InputOutputMode=XML"
     var B = "InputURL=FlashTools_Input.xml"
     var C = "OutputURL=SaveXML.php"
     var D = "PhotometricURL=PhotometricDB.php"
     var E = "InstanceKey=Any_Instance_Key_123"
     var popurl = "FlashTool.htm?" + A + "&"  + B + "&"  + C + "&"  + D + "&"  + E
     winpops=window.open(popurl,"FlashTool","width=650,height=500,status=no,resizable=yes")
}
</script>
```

The JavaScript function only needs to be defined once per page, and must include the URL to the FlashTool.htm page which embeds the FlashTool.swf movie. All URLs can be expressed as absolute or relative URLs. Relative URLs are used in this example (all the scripts and input files are in the same folder as the SWF).

A brief overview of the variables passed to FlashTool.swf via the FlashTool.htm URL.

InputOutputMode=XML — If you're reading this, use XML mode.

InputURL=FlashTools_Input.xml — The URL to the input XML file or to a script that returns the XML input as a response.

OutputURL=SaveXML.php — The URL to the script that receives and handles FlashTool output.

PhotometricURL=PhotometricDB.php — The URL to the script that receives an IES file name and/or the Instance Key and returns the following:

IES File Name	(required)
IES File Path	(either the File Path or the File Data is required)
IES File Data	(either the File Path or the File Data is required)
ImageURL	(optional)

Note: You do not need to use the PhotometericURL if you specify all the required IES data using the InputURL.

InstanceKey=Any_Instance_Key_123 — The Instance Key is passed with every XML communication to and from the server. It is simply a string which helps the server identify the instance of the FlashTool that it is communicating with. You can use it any way you wish (or not at all).

(2) JavaScript link (or equivalent) in the link tag.

This link triggers the `openFlashTool()` function defined in step (1).

```
<a href="javascript:openFlashTool()">Calculate with this awesome FlashTool</a>
```

The input for each instance of the FlashTool will vary. Usually, it only varies by which IES files it loads.
To pass the openFlashTool() function a list of IES file names, the link might look like this:

```
<a href="javascript:openFlashTool(filenames)">Calculate with this awesome FlashTool</a>
```

where:

```
filenames = "zlf1.ies, zlf2.ies, zl-ind1.ies, zl-ind2.ies"
```

requiring a change to the JavaScript function in part (1):

```
function openFlashTool_XML(list_of_filenames) {
     // do stuff with list_of_filenames
}
```

The list of file names can be:

(a) Referenced while creating the `<IES_Files>` node of the input XML file.
OR
(b) Passed in the Instance Key to be referenced by the `PhotometricURL` script.

(3) Popup HTML page where the FlashTool is embedded.

`FlashTool.htm` is the webpage that embeds the Flash movie `FlashTool.swf`.
It is best (though not required) that both files reside in the same folder on the server.

Part B -- The XML Interface

Sample Files: FlashTools_Input.xml

See the sample XML file. It contains more specific documentation about each field where necessary.

The XML specification used by the FlashTools is the same for both input and output, that is, FlashTool output can be used as input. They are identical. However, some fields are not applicable on input and are ignored by FlashTool when specified.

The high-level XML specification is shown below. You may not need all the sections, depending on which FlashTools you are implementing.

```
<?xml version="1.0" encoding="utf-8"?>                          Required with all XML communications to/from the FlashTool.
<LightingAnalysts_FlashTOOLS InstanceKey="AnyInstanceKey">      Required with all XML communications to/from the FlashTool.

     <IES_Files CalculateFileName="filename.ies">               See the  Specifying Photometry section in this document.
          <IES_File>
          <IES_File>
          <IES_File>
     </IES_Files>

     <FlashIndoor>
          <ToolSettings>                                        See the Tool Settings section in this document.
          <Zone>                                                See the sample XML file.
          <Colors>                                              See the sample XML file.
     </FlashIndoor>

     <FlashOutdoor>
          <ToolSettings>                                        See the Tool Settings section in this document.
          <Zone>                                                See the sample XML file.
          <Colors>                                              See the sample XML file.
     </FlashOutdoor>

</LightingAnalysts_FlashTOOLS>
```

Specifying Photometry

There are three ways to supply photometric data to the FlashTool. Eventually, all data must follow the XML specification shown below.

```
<IES_Files CalculateFileName="filename.ies">      Specify which of the IES files you want the FlashTool to calculate.
    <IES_File>                                    Specify one or more IES_File nodes (one for each IES file).
        <FileName></FileName>                     Required.
        <FilePath></FilePath>                     Optional:  Either the FilePath or the FileData must be specified.
        <FileData></FileData>                     Optional:  Either the FilePath or the FileData must be specified.
        <ImageURL></ImageURL>                     Optional.
    </IES_File>
</IES_Files>
```

Using CDATA

Because file names, paths, and contents may contain reserved (illegal) XML characters   & ' " > <   it is best to enclose these fields in the CDATA construct. Use CDATA to allow these characters. The only illegal string in CDATA is the three character closing tag:]]>   Beware: If the IES file data contains the string   ]]>   things could go terribly wrong. Consult your XML specification for more information.

How to specify photometry -- Option A

The Easiest Option

Use this option when the IES files reside on the server.

Use this option when you specify the photometry in the input XML file.

```
<IES_Files CalculateFileName="filename.ies">
    <IES_File>                                                         One or more IES files.
        <FileName><![CDATA[filename.ies]]></FileName>                  The name of the IES file.
        <FilePath><![CDATA[IES_Files/]]></FilePath>                    The absolute or relative URL to the folder containing the IES file.
        <FileData></FileData>                                          Not required since the FilePath is specified.
        <ImageURL><![CDATA[IES_Images/filename.jpg]]></ImageURL>       Optional.
    </IES_File>
</IES_Files>
```

How to specify photometry -- Option B

The Slowest Option

Use this option when the IES files reside in a database.

Use this option when you specify the photometry in the input XML file.

```
<IES_Files CalculateFileName="filename.ies">
    <IES_File>                                                          One or more IES files.
        <FileName><![CDATA[filename.ies]]></FileName>                   The name of the IES file.
        <FilePath></FilePath>                                           Not required since the FileData is specified.
        <FileData>                                                      The raw contents of the IES file.
            <![CDATA[
                IESNA91
                [XML ILLEGAL CHARACTERS]  <  >  &  '  "
                [TEST]12345
                [DATE] 19-FEB-97
                [MANUFAC]Z-LUX... FOR THIS PLANET AND BEYOND
                [LUMCAT]XML{  <  >  &  '  "  }
                [LUMINAIRE]Common Asteriod series - 8 CELL PARABOLIC 1x4
                [LAMP]1 LAMP (F40T12/CW) RATED 3150 LUMENS
                [DISCLAIMER]THIS DATA IS FICTITIOUS AND SHOULD BE USED FOR
                [MORE]AGI32 DEMONSTRATION ONLY
                [MORE]LIGHTING ANALYSTS IS NOT RESPONSIBLE FOR MISAPPLICATION OF THIS DATA
                TILT=NONE
                1  3150  1  19  5  1  1  1  4  0
                1 1 53
                0  5  10  15  20  25  30  35  40  45  50  55  60  65  70  75  80  85  90
                0  22.5  45  67.5  90
                750  767  760  725  695  650  620  585  530  485  420  340  265  135  20  5  1  0  0
                750  770  765  730  705  670  640  615  575  550  515  480  405  220  30  10  2  0  0
                750  770  775  750  730  700  700  730  780  805  750  605  395  180  65  10  5  0  0
                750  770  780  760  750  750  820  920  970  940  775  515  290  180  65  15  5  1  0
                750  770  780  765  770  780  880  985  1035  985  790  450  300  195  60  15  5  1  0
            ]]>
        </FileData>
        <ImageURL><![CDATA[IES_Images/filename.jpg]]></ImageURL>         Optional.
    </IES_File>
</IES_Files>
```

How to specify photometry -- Option C

The Best , Most Efficient Option

Use this option to employ the PhotometricURL in addition to the input XML file (InputURL). Since only the FileName is specified in the input XML file, this option can lead to faster loading times and reduced bandwidth, especially in comparison with Option B or where many IES files are to be loaded into the FlashTool (remember, it is likely that only some of the IES files will be selected by the user).

(1) The input XML file is created as shown.

```
<IES_Files CalculateFileName="filename.ies">
    <IES_File>                                                One or more IES files.
        <FileName><![CDATA[filename.ies]]></FileName>         The name of the IES file.
    </IES_File>
</IES_Files>
```

(2) The user selects the IES file (from the drop-down list).

(3) The FlashTool checks for `FileData` and for a `FilePath`.

(4) If neither was supplied, the FlashTool sends the following XML to the Photometric URL.

```
<?xml version="1.0" encoding="utf-8"?>
<LightingAnalysts_FlashTOOLS InstanceKey="AnyInstanceKey">
    <IES_Files CalculateFileName="filename.ies">
        <IES_File>
            <FileName><![CDATA[filename.ies]]></FileName>
            <FilePath></FilePath>
            <FileData></FileData>
            <ImageURL></ImageURL>
        </IES_File>
    </IES_Files>
<LightingAnalysts_FlashTOOLS>
```

(5) The Photometric URL (script) returns the XML response for the IES file as shown in Option A or Option B.

Note: All XML communication with the FlashTool must include the XML version/encoding node and the root node (Lighting Analysts node).

Tool Settings

Images

Allows you to specify graphics to be loaded into the FlashTool. Larger images will increase the load time (download time) of the FlashTool.

`<AllowLumianireImages>True</AllowLumianireImages>` Setting this to `False` hides the *Show Luminaire Image* button in the *Luminaires* section of the FlashTool.

Custom Commands

You can extend the functionality of the FlashTool by adding custom commands that point to URLs or scripts on your server. You can specify images for the command button and where the button appears in the FlashTool.

There are several command types available.

(a) `URL_Open` Opens the given URL in the current browser window.
(a) `URL_Open_Blank` Opens the given URL in a new browser window.
(b) `Script_MinimalOutput` POSTs minimal XML output to a server script, which can 'do stuff'.
(c) `Script_FullOutput` POSTs full XML output to a server script, which can 'do stuff'.
(d) `Script_MinimalOutput_FullInput` In addition to (b), the script must return full XML input which reinitializes the FlashTool with new data.
(e) `Script_FullOutput_FullInput` In addition to (c), the script must return full XML input which reinitializes the FlashTool with new data.
(f) `URL_Open_Blank_FullOutput` Opens the given URL in a new browser window and POSTs the full XML output to the given URL.

Minimal Output: Sends the root node (only) to a server script. The root node, of course, includes the Instance Key.

Full Output: Sends the entire XML specification of the current state of the FlashTool. Use full output when you intend to exactly restore or duplicate an instance of the FlashTool or for tracking/logging purposes.

Full input: Returns an entire XML specification to the FlashTool (exactly the same specification as the initial XML input file). The FlashTool is reloaded with the new data. Use full input when you want to restore the original state of the FlashTool or when you want to completely change the state of the FlashTool.

Partial input : It is possible to specify partial input, such as specifying only the `ToolSettings` *node or other variable nodes individually. In this case, do not include empty nodes, that is, only include the nodes and variable nodes you wish to specify.*

<u>Note</u> : Full Input is still required when the FlashTool is opened (via the `InputURL` *).*

`HelpText` Displays the given text in the Information/Help section or ToolTip when the mouse hovers over the command button.

Custom Commands (continued)

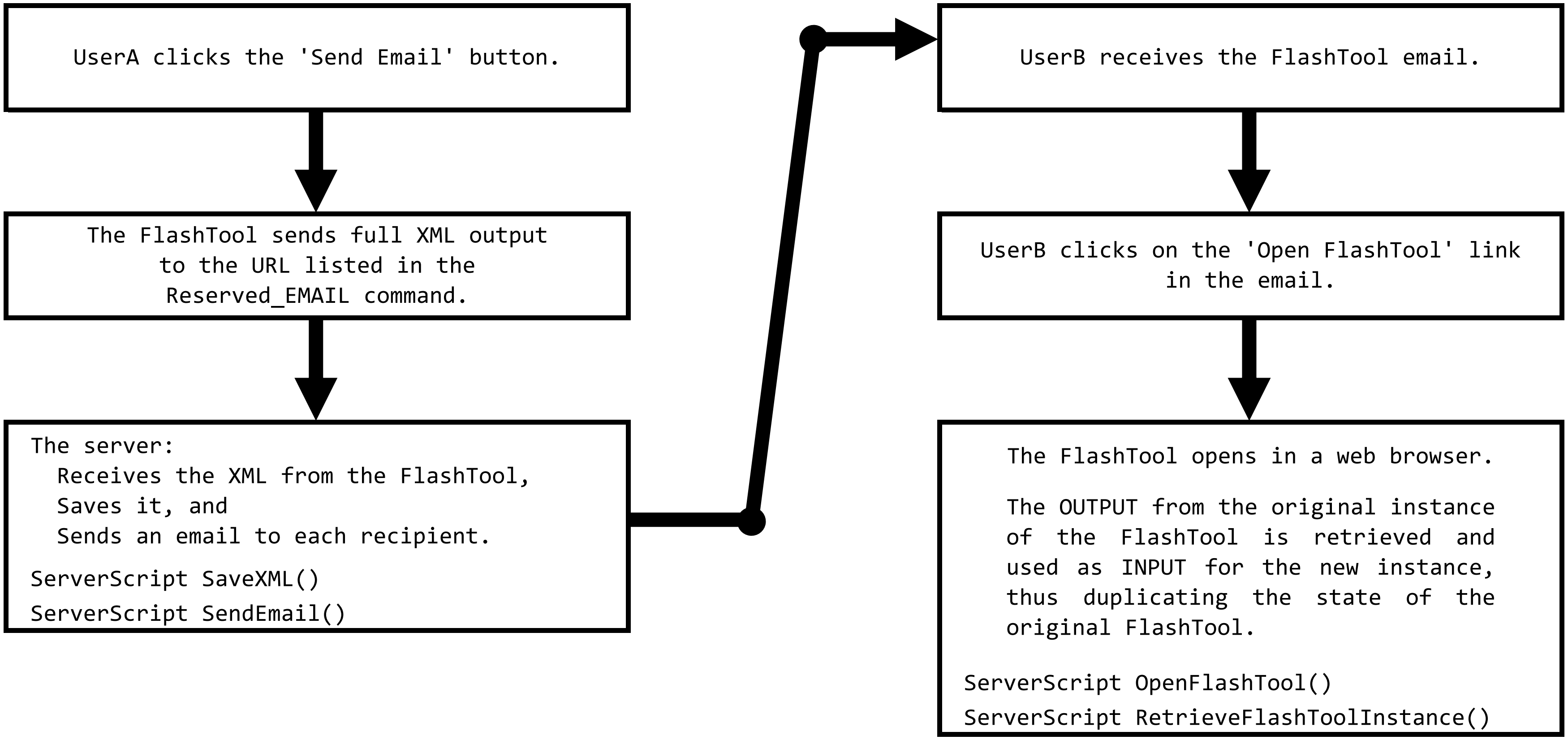
Certain commands are reserved for dedicated use within the tool (the user interface is already built-in). However, you must customize the TYPE and URL to integrate with your website. All reserved commands begin with the prefix "Reserved".

```
<Command UniqueName="Reserved_ERROR">          Reserved for the 'Click here to notify us' button on the error screen.
     <Type>URL_Open_Blank</Type>
     <URL>http://www.agi32.com/forums/</URL>
</Command>
<Command UniqueName="Reserved_HELP">           Reserved for the 'Help' button in the Output/Help section.
     <Type>URL_Open_Blank</Type>
     <URL>http://www.agi32.com/forums/</URL>
</Command>
<Command UniqueName="Reserved_EMAIL">          Reserved for the 'Send Email' button in the Output/Help section.
     <Type>Script_FullOutput</Type>
     <URL>Email.php</URL>
</Command>
```

Email

You can add email functionality with server-side scripting. If you decide not to implement email functionality, make sure to hide the Email controls.

`<AllowEmailOutput>True</AllowEmailOutput>` Setting this to `False` hides the *Email* controls in the *Output* section of the FlashTool.



UserA clicks the 'Send Email' button.

The FlashTool sends full XML output to the URL listed in the Reserved_EMAIL command.

The server:
  Receives the XML from the FlashTool,
  Saves it, and
  Sends an email to each recipient.

```
ServerScript SaveXML()
ServerScript SendEmail()
```

UserB receives the FlashTool email.

UserB clicks on the 'Open FlashTool' link in the email.

The FlashTool opens in a web browser.

The OUTPUT from the original instance of the FlashTool is retrieved and used as INPUT for the new instance, thus duplicating the state of the original FlashTool.

```
ServerScript OpenFlashTool()
ServerScript RetrieveFlashToolInstance()
```

Email (continued)

```
ServerScript SaveXML()
     Receive XML data from the FlashTool.
     Generate a unique key.
     Save the XML data to the server (file or database).
     Associate the XML data with the unique key so that, given the key, the XML data can be retrieved.

     Call ServerScript SendEmail(Key, List)
End ServerScript


ServerScript SendEmail(Key, List)
     Generate email content as desired.
     Include a 'Click Here' link that points to ServerScript OpenFlashTool()
          Example:  http://www.domain.com/RetrieveFlashToolInstance.php?Key=UniqueID_1234567890
End ServerScript


ServerScript OpenFlashTool()
     Receive the KEY variable passed to this script in the URL.
     Open the FlashTool in a web page (refer to Part A of this document).
          Set the InstanceKey variable (of the FlashTool) to the KEY variable that was passed to this script.
          Set the InputURL (of the FlashTool) to point to ServerScript RetrieveFlashToolInstance()
End ServerScript


ServerScript RetrieveFlashToolInstance()
     Receive new XML data from the FlashTool.
     Parse the InstanceKey attribute from the root node of the XML data.
     Use the InstanceKey to lookup and retrieve the saved XML data.
     Return the saved XML data to the FlashTool to be used as input.
End ServerScript
```